システムLSI 検証技術入門

デバイスの記事

第8回

論理回路の等価性検証技術（3）

――LSI設計フローと等価性検証の関係

藤田昌宏

今回は，フォーマル・ベリフィケーションの一つである等価性検証（equivalence checking）技術のうち，組み合わせ回路を対象とする技術を総括する．回路最適化を含むRTLからの設計手法の詳細と，これらに等価性検証を適用する場合の具体的な処理手順について解説する．また，大規模LSIや高性能LSIの設計で頻繁に利用されている「リタイミング」を実施した際の等価性検証技術についても説明する．（編集部）

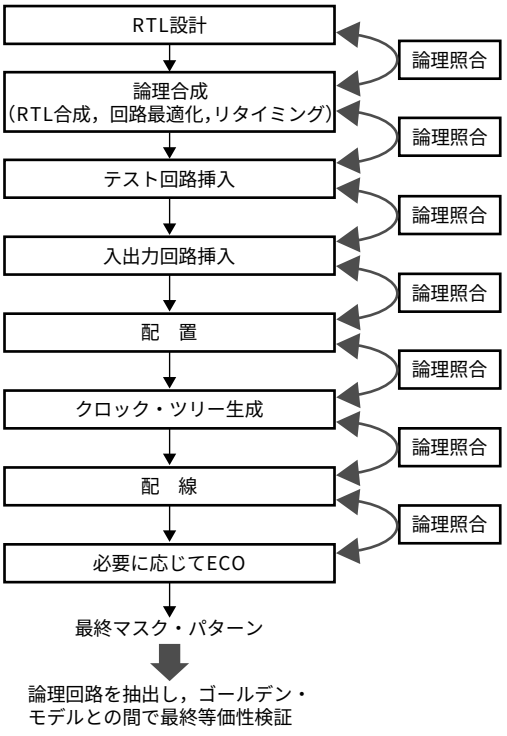

図1　現状の設計の流れと等価性検証の適用範囲

RTLの抽象度で論理機能を設計し，論理合成処理を施す．そのあと，製造故障テストに必要なテスト回路の挿入（DFT：design for testability）を行う．そして，必要に応じて入出力回路（I/Oセル）を挿入し，配置処理，クロック・ツリー作成，配線処理へと進む．最後に，必要に応じてECO（engineering change order）などの仕様変更や論理変更の処理が行われる．

今回は，二つの設計データの論理的等価性を判定する「等価性検証（equivalence checking）」のうち，組み合わせ回路を対象とする技術の総括を行います．まず初めに，RTL（register transfer level；レジスタ転送レベル）の設計手法をふりかえり，回路最適化と等価性検証の関係について説明します．次に，リタイミング手法（順序回路の最適化技術の一つ）によって最適化処理が施された回路の等価性検証手法について説明します．最後に，現在，主要なEDAベンダから提供されている等価性検証ツールの概略を述べ，それらの利用上の問題点や注意点をまとめます．

● RTLからの論理回路設計と等価性検証をおさらい

以前にも説明しましたが，もう一度，RTL以降の設計の流れを**図1**に示します．

まず，設計対象が満たすべき論理や状態遷移をRTLの抽象度で記述します．そして，一種のコンパイラ（論理合成ツール）を使って，このRTL記述から各フリップフロップの次の値を決める論理式を生成します（RTL合成）．このとき，状態割り付けなどを行う「順序回路最適化処理」と，フリップフロップ部分を取り除いた純粋な組み合わせ回路部分のみを対象とする「組み合わせ回路最適化処理」の2種類が施されます．タイミング（遅延）調整も含めて，ひととおり最適化された後，テスト回路の挿入や入出力回路（I/Oセル）の挿入を行い，続いて回路全体のおおざっぱな配置を決定します．全体の配置が決まったら，クロック・ツリー（クロック木）を生成し，クロック信号の伝播に問題がないことを確認しながら，残りの信号線の配線を行っていきます．最後のECO（engineering change order）の工程では，主にタイミング調整のために，配線や配置，場合によってはRTL記述を部分的に変更します．ECOを部分的な変更

にとどめる理由は，それまでに行ったタイミング最適化などの結果（回路やレイアウト）をできるだけ生かしたいからです．

図1にも示したように，等価性検証では，設計の各工程の処理に対して，その処理の前の回路（設計データ）と後の回路の論理的な等価性をチェックすることになります．**図1**のうち，テスト回路の挿入については，挿入したテスト回路をなんらかの方法で除去し，除去したものと元の回路の等価性をチェックします．したがって，検証ツール側は，一般的なテスト用スキャン回路のスキャン入力やスキャン出力，スキャンの制御信号線の情報を受け取り，それに基づいてスキャン回路を除去するようになっています．ただし，スキャン回路が特殊な方式の場合，検証ツール側で自動的に除去できず，検証不能になることがあります．

テスト回路挿入後は，おおむねフリップフロップがスキャン用のものに変更されていたり，一つまたは複数のスキャン・チェーン（スキャン・フリップフロップを数珠つなぎにして構成したシフト・レジスタ）が挿入されていたりします．設計者がこれらの情報を検証ツールに直接指示しなければならないケースも少なくありません．また，製造故障テスト用にテスト・ポイントが挿入されている場合もあります．その際には，設計者がテスト・ポイントを挿入した箇所を指示して除去する必要があります．

ここは，自分たちが利用しているテスト回路を自動的に除去できるようにツールを調整したくなる部分ですが，EDAベンダの提供するツールがかならずしもこのようなカスタマイズに対応しているとは限りません．一方，テスト回路が市販のEDAツールによって自動生成されている場合，検証ツール側でもテスト回路の方式がわかっているので，自動的に除去できる場合が多いようです．

入出力回路の挿入についても同じことが言えます．単純なバッファだけであれば信号線の対応の情報を与えてやるだけですみます．しかし，複数の信号が一つのピンを共有するような場合には，そのことを設計者がツールに指示してやらなければなりません．

続く配置処理については，論理回路は原則として変更されないので，単純な等価性検証になります．

クロック・ツリー生成については，通常の同期回路であれば，等価性検証に影響することはありません．しかし，ゲーテッド・クロック（gated clock）回路などを利用している場合は，そのことを明示する必要があります．最近は，低消費電力化を意識してLSIを設計するのが普通で，クロックをあの手この手で制御している場合も少なくありません．そのため，設計者が直接指示しないといけない部分が増えています．また，多数のクロックを使っている場合や，それらを多相クロックとして利用している場合，指示が必要になるケースがあります．例えば，ある組み合わせ回路部分について，その入力が同期しているクロックとその出力が同期しているクロックが異なる場合などです．

● RTL記述を含む等価性判定は数千ゲート程度まで

RTLデータから論理式を生成するRTL合成と回路最適化処理に対する等価性検証については，**図2**を見てください．HDL（hardware description language；ハードウェア記述言語）で表現された設計記述に対して，自動合成や人手設計が適用されます．また，自動合成でも主にデータパス系のブロックを扱うモジュール合成などにより，ネットリストやマスク・パターンが生成されることもあります．

まず**図2**の中で，通常，論理合成と呼ばれる流れについての等価性検証を考えます．処理手順としては，まず，HDLで表現されたRTLの記述をコンパイルし，そこから論理式を生成するRTL合成の処理を行います．このRTL合成では，内部で状態割り付けやレジスタの生成などが行

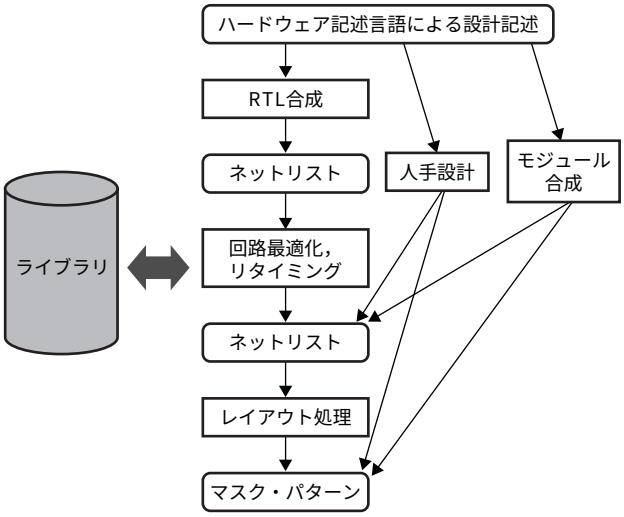

図2　RTLからの論理設計の流れ

まず，HDL（hardware description language）によって表現されたRTLの記述から論理式を生成する（この処理は「RTL 合成」と呼ばれる）．次に，得られた論理式（論理回路）に対して，論理回路レベル（ゲート・レベル）の最適化が施される．最後に配置配線処理を行い，ネットリストからマスク・パターンを生成する．人手設計によってRTL記述からネットリストを生成する場合やモジュール合成が行われる場合もある．