

プロセッサ設計における STA と SSTA の解析結果比較および分析

伊藤 則之 小松 裕成 杉山 広行 備前 直美 井口 克己 吉田 裕司

富士通株式会社 〒211-8588 川崎市中原区上小田中 4-1-1

E-mail: {ito.noriyuki, komat, sugiyamax, bizen, iguchi.katsumi, yuji.yoshida}@jp.fujitsu.com

あらまし 半導体プロセスに微細化により、製造ばらつきが性能に与える影響が大きくなった。そのため、従来のタイミング解析 (STA) でのワーストケース設計ではタイミング設計は困難となっており、新たに統計的タイミング解析 (SSTA) の手法が導入され始めた。本論文では、プロセッサの設計において実際に SSTA を適用し、クリティカルパスでの従来の STA での解析結果との比較・分析を行った。

キーワード 統計的タイミング解析, プロセッサ, クリティカルパス

Comparison between STA and SSTA Results in Microprocessor Design

Noriyuki ITO Hiroaki KOMATSU Hiroyuki SUGIYAMA Naomi BIZEN

Katsumi IGUCHI and Yuji YOSHIDA

Fujitsu Limited 4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki, 211-8588, Japan

E-mail: {ito.noriyuki, komat, sugiyamax, bizen, iguchi.katsumi, yuji.yoshida}@jp.fujitsu.com

Abstract Due to decreasing feature sizes, process variation influences performance. Therefore, timing design by worst-case design based on traditional timing analysis (STA) is impractical. A new statistical timing analysis is introduced. In this paper, we compared between STA and SSTA results in microprocessor design.

Keyword Statistical timing analysis, Microprocessor, Critical Path

1. はじめに

従来の LSI 設計では、製造ばらつきによる最悪のディレイを考慮したワーストケース設計が行われ、製造したチップはすべて目標の周波数で動作するようなタイミング設計手法が一般的であった。しかし、微細化に伴ってばらつきが大きくなるにつれて、この手法は過剰なタイミングマージンを含む設計となっている。このため、高性能な LSI 設計において従来の設計手法を用いたのでは、限られた開発期間の中で目標性能を満たす設計は非常に難しくなってきた。そこで、より現実的な設計を可能にするために、新たな統計的タイミング解析 (SSTA) という手法が研究されてきた。これは、パスのディレイ値を求める際に、パス上のセルのディレイの最悪値を積算して行く従来手法とは異なり、確率分布で表現されたセルのディレイを統計的に積算する。これまでの SSTA の研究は、モデル化、アルゴリズム、精度、速度などに関するものが多く、実際の設計にどのように適用するかについての議論や実際に適用された事例の報告はまだ多くない。文献[1,2]では、STA を SSTA で置き換えて設計フローを構築している。文献[3]では、STA で検出されたクリティカルパスに対して、パススペースの SSTA を適用して、周波数歩留まりに対する影響を解析するという手法が採られている。文献[4]では、STA でどの程度のばらつきを

考慮すると目的の周波数歩留まりを得ることができるか、これを決めるために SSTA を適用している。これら 3 つの適用形態に対して、筆者らの SSTA 適用の形態は、 3σ より少ないばらつきで STA を使ったタイミング設計を行い、SSTA で周波数歩留まり設計を行うものである。つまり、筆者らの設計フローでは、STA で目標周波数をチェックし、SSTA で目標周波数歩留まりをチェックする。

本稿では、第 2 章でプロセッサ設計におけるタイミング設計についてその概要を説明する。第 3 章では、SSTA の実現方法および設計への適用フローについて説明する。第 4 章では、SSTA を使った実際の設計改善の適用事例を説明する。第 5 章では、SSTA と STA の解析結果を比較して、クリティカルパスの順位が異なるケースについて、その原因を分析する。その後、第 6 章で STA と SSTA の実行時間の比較を行い、第 7 章で全体のまとめを述べる。

2. プロセッサのタイミング設計

筆者らのプロセッサ設計では、GHz レベルの高い目標周波数性能を実現するため、スタンダードセルレベルでのセミカスタム設計とトランジスタレベルでのカスタム設計を適用している (以降、セミカスタム設計とカスタム設計をまとめてカスタム設計と

呼ぶ)。論理設計では直接ゲートレベルのネットリスト記述が行われ、RTL 記述後に論理合成を行ってゲートレベルに変換する手法は使われていない。理由は、RTL 記述の改善や論理合成の利用技術だけで目標性能を実現できるネットリストを作成する保証がないためである。目標性能が実現できていない場合、論理合成後のネットリストに設計者が手を加えてタイミング改善を行うことは現実的でない。これは、論理合成後のネットリストは人間が論理を読むことを想定していないためである。筆者らのネットリスト設計環境では、自社で定義した生産性の高いネットリスト記述言語 ADL を使って設計者は論理設計を行う。この ADL 言語は、たとえば 64 ビット分など複数のセルを一行で記述でき、またインスタンス名の中で図面のページおよびページ内の場所を一緒に記述することができる。この言語記述は論理設計の原本となり、変換ツールによって verilog などに変換され、図面印刷その他のツールの入力となる。

ゲートレベルでのネットリスト記述による論理設計では、設計者が論理を読む上で論理図面が非常に重要となる。論理図面では、各インスタンスは指定されたページ名とページ内ロケーションに配置される。各インスタンスが持つこのページ名とページ内ロケーションは、論理ネットリスト作成の際、設計者は各インスタンス名として論理図面のページ名とページ内ロケーションの組み合わせたものを指示することによりネットリストに記述される。論理図面出力ツールは、入力されたネットリストのインスタンス名から論理図面情報を抽出して図面印刷を行うことができる。

論理図面は論理設計のときだけでなく、タイミング設計を行う段階でも非常に重要な役割を示す。タイミング改善の段階で、どの部分のディレイが厳しいのか、ネットの配線長は長いのか短いのかなど、論理図面上にタイミング情報やレイアウト情報を一緒に表示することによって、論理図面を見ながらタイミング改善を行うことが可能となる。そのため、タイミング解析ツールからのスラック情報、またレイアウトツールからのネットの配線長情報を論理図面印刷ツールに入力して、これら情報を一緒に表示することが可能となっている。

論理設計後のレイアウトやタイミング解析のために、チップは複数のサブチップに分割され、各サブチップはさらに複数のブロックに分割される[5]。レイアウトはブロック単位で行われた後、サブチップやチップへと組み上げられる。タイミング解析はサブチップ単位で毎日実行され、その結果に基づいてタイミング改善を行う。チップ全体のタイミング解析も数日間隔で定期的に行われる。設計初期から設計完了までの

タイミング設計の流れは図 1 に示される通りである。設計初期から中期にかけて、セル配置レベルでの性能改善を行い、この段階では、配線ディレイはスタイナー木で評価し、クロックディレイは 0 としてタイミング解析を行う。この解析をもとに、論理やセル配置を設計者が改善しながら、目標ディレイに近づけて行く。目標ディレイに近づいてくるとラッチの配置位置が確定してくるので、クロック回路を設計して、配線を含めてレイアウトを行う。この段階では、実際のクロック配線に基づくディレイ計算によりクロックスキューを考慮したタイミング解析に入る。ここでのタイミング最適化がほぼ終了すると、すべての信号について実際に配線を行い、実配線ベースでのタイミング解析を行う。チップ全体ですべてのパスが目標のディレイ以下になってきた段階で、タイミングやレイアウトの最終検証を行う。図 2 に示されるグラフは、横軸がパスディレイ値で縦軸がパス数となるチップ全体のパスディレイ分布である。この分布が時系列に管理され、チップ全体のタイミング改善が管理される[6]。

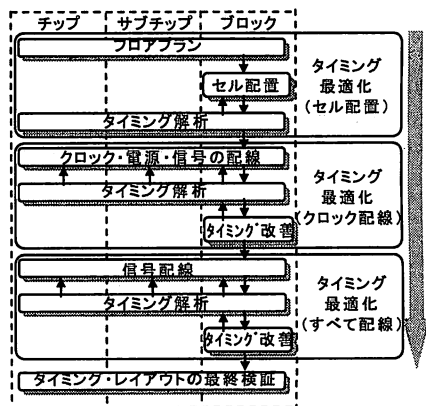


図 1. タイミング解析フロー

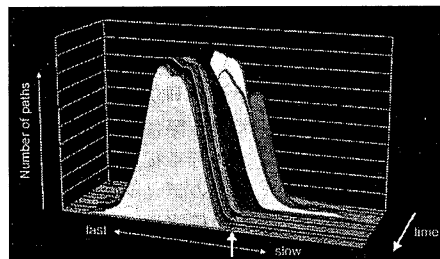


図 2. STA によるディレイ分布の改善[6]

3. SSTA の実現方法と運用フロー

筆者らはプロセッサ設計で使っている自社開発の STA ツールに対して、パスディレイ計算の際のディレイ

イ値積算と MAX 計算の2つの部分をそれぞれ統計的積算と統計的 MAX 処理に置き換えることによりブロックベースの SSTA ツールを開発した。ブロックベースの適用は、図3のように、始点と終点のラッチペアの間の論理とそれらラッチに供給されるクロック回路を含む回路部分に対してとなる。SSTA を適用する始点・終点ラッチペアは、STA が出力するクリティカルパスリストにおいてクリティカル度が高い複数のパスとする。STA は論理上考えられるすべての始点・終点ラッチペアに対してパスディレイ計算を行うが、SSTA では処理時間が STA に比べて大きいため、チップの周波数歩留まりに影響するクリティカルパスに限定して SSTA を適用する。チップの周波数歩留まりに影響するクリティカルパスの決め方は、文献[7]に示される。各クリティカルパスに SSTA を適用して得られたディレイの確率密度分布を、SSTA を適用したクリティカルパス数分だけ統計的 MAX 処理を行うことにより、図4に示されるように SSTA 結果が得られる。結果は PDF (Probability Density Function) 表現から CDF (Cumulative Distribution Function) 表現に変換することにより、チップ全体の周波数歩留まり分布が得られる。なお、プロセッサは一般の ASIC に比べるとタイミング性能制約が厳しいため、パスの論理段数が少なく、パスによる論理段数の差が小さい。従って、チップで一番クリティカルなパスのディレイ値付近の値を持つパスは多数存在する。

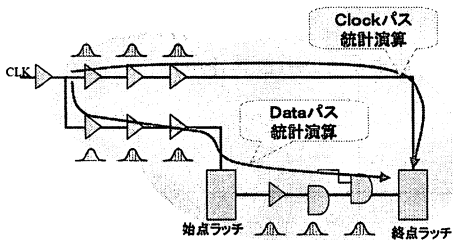


図3. SSTA の適用単位

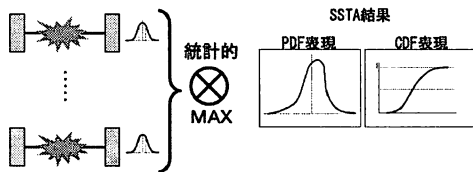


図4. チップ全体の周波数歩留まり計算

SSTA を実際のプロセッサの設計フローに適用する際、まず判断する必要のある重要な事項は、従来のタイミング解析を SSTA で置き換えるのか否かということである。筆者らのプロセッサ設計において、SSTA

は従来の STA を置き換えるのではなく、新たな追加されたツールという位置付けで運用を行う。すなわち、まず従来の STA ツールを使って従来と同様のディレイ目標をまずクリアするように設計する。このとき、製造ばらつきを従来のように 3σ を考慮したワースト設計は現実的でないため、製造ばらつき範囲 $\pm n\sigma$ の n の値を緩和する。この STA によるパスディレイのチェックをクリアした後で SSTA を実行して、SSTA 用に新たに設定された周波数歩留まり目標をクリアするように設計を改善する。もし、周波数歩留まり目標をクリアしていない場合には、クリアするための STA に対する新たなディレイ目標がツールにより算出される。算出の概要は算出のために、まず STA のためのより厳しいディレイ目標を仮に決めて、このディレイ値を越えているすべてのパスがこの仮のディレイ目標をクリアしたことを想定して周波数歩留まり分布を計算する。この分布において、目標の周波数での歩留まりが目標値と同じになるように、目標のディレイ値を変化させながら検索する。図5において、STA の現目標をクリアしても周波数歩留まりは目標に達していないため、目標を Xps だけ厳しくした新目標を決め、STA でその目標をクリアする。この結果、現目標での周波数歩留まりが目標に到達する。この詳細は文献[8]に示される。

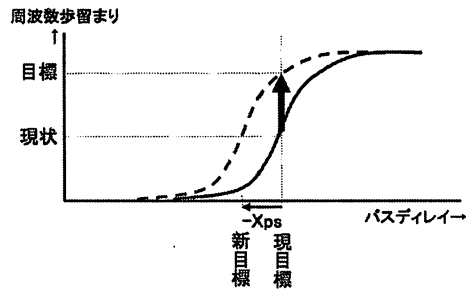


図5. SSTA を利用した新たな STA 目標の設定

4. SSTA を利用した設計改善

第3章で示したように、SSTA は設計対象の LSI に対して、周波数歩留まり分布を計算することができる。プロセッサの設計者は、この SSTA 結果を見て、目標の周波数歩留まりを達成するために、STA のための新たなディレイ目標に向かって設計改善を行う。しかし、このディレイ目標が限られた設計期間で達成できる保証はない。このような状況では、チップ全体の周波数歩留まりに一番影響を持っている部分を順次改善して行くことがより重要である。

そこで、筆者らはチップ全体の周波数歩留まりに一番影響を持っている部分を特定するために、まずチップを構成するサブチップ単位の周波数歩留まり分布を全サブチップについて計算して、それらを重ねて表示

することとした。具体的には、チップ全体の周波数歩留りに影響を持つ10万パスについて、そこから各サブチップのパスに分類して、同一のサブチップのパス全体に対して統計的 MAX 演算を適用してサブチップ単位の周波数歩留り分布を計算した。なお、サブチップ間パスはすべて1つにまとめて周波数歩留りを計算している。現在開発中の実際のプロセッサ設計に適用した結果が図6に示される。あるディレイのチェックポイントにおいて（このチェックポイントは説明の都合上、仮に設定したディレイ値のポイントである）、周波数歩留りが一番悪いサブチップCでは、その周波数歩留りはチップ全体の周波数歩留り74%に非常に近い値となっている。これは、サブチップCがチップ全体の周波数歩留りに一番影響を与えていることを意味している。このサブチップCは、実チップの周波数選別でも、これまで周波数歩留りに一番影響を持つ部分として知られており、その事実と一致していた。サブチップ間パスも通常ディレイ的には厳しいところであり、図6ではサブチップクロスと示されるものがその周波数歩留り分布となり、その厳しさの度合いを知ることができる。

このように、サブチップ毎の周波数歩留り分布を計算することにより、チップ全体の周波数歩留りを上げるためには、どのサブチップのディレイ改善が一番重要であるかを定量的に把握することができる。設計者は、このサブチップ単位の周波数歩留り分布を見ながら、サブチップでの周波数歩留り改善のためにパスのディレイ改善を行ってゆく。図7は、サブチップCにおいて重点的にパスのディレイ改善を行って周波数歩留りを改善した途中の結果である。図6と同じディレイのチェックポイントにおいて、チップ全体の周波数歩留りは89%まで改善しているが、依然としてサブチップCがチップ全体の周波数歩留りに一番影響を与えていることがわかる。このサブチップCおよびその他のサブチップでもさらにパスのディレイ改善を行った最終的な周波数歩留り分布の結果が図8である。図6と同じディレイのチェックポイントにおいて、チップ全体の周波数歩留りは91%まで改善している。図7にはあったサブチップのグラフのいくつかは図8ではなくなっているものがある。この理由は、サブチップC以外のサブチップでもディレイ改善を行った結果、チップ全体の周波数歩留りに影響を持つ10万パスの中にそれらサブチップ内のパスは現れてきていないためである。第3章では、あらかじめ設定された周波数歩留り目標を達成するためにSSTAを適用するというを示したが、ここで示したようにサブチップなどチップを構成する設計単位毎に周波数歩留り分布を計算して比較することにより、

周波数歩留りを改善すべきところを的確に知ることができるようになる。

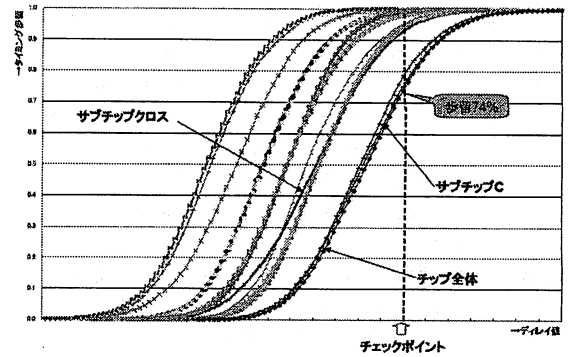


図6. 周波数歩留り（改善前）

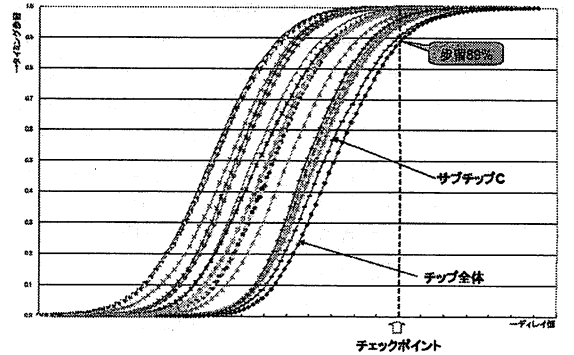


図7. 周波数歩留り（改善途中）

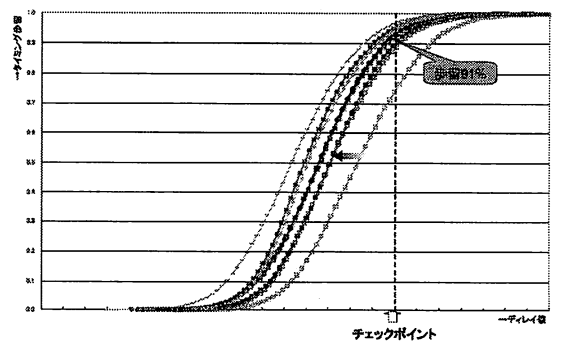


図8. 周波数歩留り（改善後）

5. STA と SSTA でのパスの順位比較

筆者らは自社で開発した STA と SSTA を両方持っており、しかも SSTA は STA をベースに開発したため、パスディレイ計算の際のディレイ値の積算部分と

MAX 処理部分以外はほぼ同じである。このような STA と SSTA を同じプロセッサ設計に適用して、それぞれが出力するクリティカルパスの順位を比較した。なお、SSTA でのクリティカルパスの順位を比較は、パスのディレイの確率密度分布の平均値を選んで比較する。また、STA では製造ばらつきを従来の 3σ より大きく緩和したものが適用されている。その結果の一部を図 9 に示す。STA の 1 番、2 番、3 番の各パスは、SSTA ではそれぞれ 139 番、59 番、785 番となっている。なお、図 9 で SubChip の欄には、STA のパスがどのサブチップのものかを示している。特に変動の大きい 16 番のパスは、SSTA では 14098 番となっている。逆に SSTA 側から見ると、SSTA で 1 番、2 番、3 番の各パスは、STA ではそれぞれ 1266 番、1359 番、1381 番となっている。特に変動の大きい 943 番のパスは、STA では 92870 番となっている。

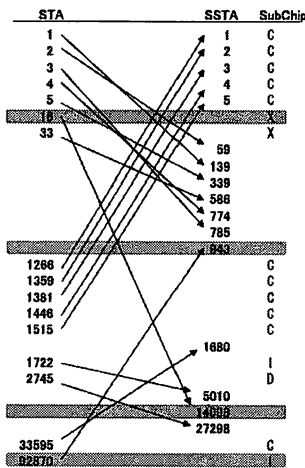


図 9. STA と SSTA でのクリティカルパスの変動

このような STA と SSTA で同じパスの順位が変動する理由を考察する。まず、STA で 1266 番、1359 番、1381 番のパスが SSTA ではそれぞれ 1 番、2 番、3 番となる理由は、パスの合流の際、STA ではその解析が SSTA より楽観的であることに起因する。図 10 の 4 入力 AND ゲートにおいて、一番上の入力の信号が他の 3 つの入力の信号よりディレイ的に遅い場合、STA では一番上の入力を通るパスのディレイに他の 3 つの入力を通るパスのディレイが影響を与えることはない。SSTA の場合、一番上の入力の信号が他の 3 つの入力の信号よりディレイ的にある一定以上に遅い場合、STA と同じく一番上の入力を通るパスのディレイに他の 3 つの入力を通るパスのディレイが影響を与えることはない。しかし、他の 3 つの入力の信号のディレイ

値が一番上の信号のディレイ値に近づいてきた場合、他の 3 つの入力を通るパスのディレイが一番上の入力を通るパスのディレイに与える影響は無視できなくなる。具体的には、図 11 において、A のグラフが、一番上の入力の信号が他の 3 つの入力の信号よりディレイ的にある一定以上に遅い場合の AND ゲート出力でのディレイ確率密度分布とする。もし、他の 3 つの入力の信号が一番上の入力の信号とほぼ同時に到着する場合の AND ゲート出力でのディレイ確率密度分布は、B のグラフのように A に比べて遅いほうにずれることがわかる。従って、パスの合流が多い場合には、SSTA では STA よりクリティカルパスの順位は遅いほうに変動する。STA の 1 番、2 番、3 番の各パスが、SSTA でそれぞれ 139 番、59 番、785 番となった理由は、パスの合流による SSTA での変動があまりないため、パスの合流により上位に変動したパスによって相対的に順位が下がったことが原因となっている。

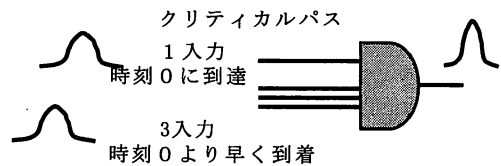


図 10. 他の入力のディレイの影響

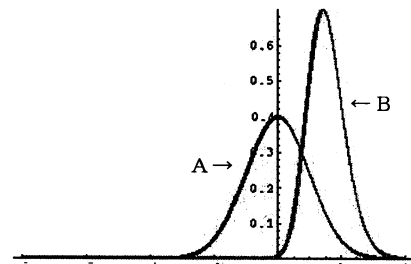


図 11. AND ゲート出力でのディレイ確率密度分布

次に、STA で 16 番および 92870 番のパスが、SSTA でそれぞれ 14098 番および 943 番と大きく変動している理由について分析する。表 1 は、これら 2 つのパス (#2,4) と先にすでに述べたパスの 2 つ (#1,3) の合計 4 つのパスについて、パスの回路構成の特徴についてまとめたものである。一般に、パスのディレイはゲートディレイの成分と配線ディレイの成分に分けることができる。表 1 に示した 4 つのパスの中で、#2 のパスは #1 のパスに比べて、パスディレイ全体に占めるゲートディレイの比率が大きくなっている。ゲートディレイの比率が大きい理由は、ゲート間の配線の長さが #1 のパ

スより短いか、配線が長くても抵抗の小さい太い配線を使っているためである。#2のパスのケースは、サブチップ内に閉じていないサブチップ間のパスであるため、配線が長くても抵抗の小さい太い配線を使っているケースに該当する。この章の冒頭ですでに述べたように、ここで適用した STA では製造ばらつきを従来の 3σ より大きく緩和したものが適用されている。このとき、緩和してはいるが完全に 0 とはしていない。製造ばらつきは、配線ディレイよりゲートディレイにより大きく影響するため、STA ではゲートディレイの比率が大きい#2のパスが#1のパスに比べてより悲観的に評価されている。また、パスの送りと受けに供給されるクロック供給パスの段数が多くゲートディレイが大きい場合も、その変動はより大きくなる。#2のパスが#1のパスより相対的に順位を大きく下げる理由は、このような STA での悲観性がより大きいためである。なお、#3のパスの SSTA での変動は、パスの合流による上位への変動のほうが STA の悲観性の排除による下位への変動より大きいため、結果として SSTA では上位に変動している。最後に、#4のパスが SSTA で#3のパスより大きく順位を上げている理由は、#4のパスでは#3のパスに比べてゲート段数が少ないことが要因の一つである。つまり1段のゲートのディレイが大きく、この大きなゲートディレイに対して同様のパスが合流するため、統計的 MAX 計算を行うことにより#3のパスのようにゲートディレイが小さい場合に比べて大きなディレイ変動となる。もう一つの要因は、#2のように SSTA で順位を下げるパスによる影響である。

表 1. パスの特徴

#	パス (STA→SSTA)	合流数	段数	パスディレイに占めるゲートディレイの比率
1	1 → 139	無	11	小
2	16 → 14098	無	11	大
3	1266 → 1	多	11	大
4	92870 → 943	多	6	大

6. STA と SSTA の実行時間比較

現在開発中のプロセッサ設計に適用した SSTA ツール実行では、クリティカルな 100,000 パス（ここでの1パスは始点と終点のラッチペアを意味しており、このラッチペアの間の論理とそれらラッチに供給されるクロック回路を含む回路部分に対してブロックベースの SSTA を適用する）に対して、CPU 時間は約 3.3 時間、使用メモリは約 35GB となった。比較のために、通常の STA 実行については、全 14M パスに対する解析は、CPU 時間は約 1 時間、使用メモリは約 20GB となっ

ている。なお、実行した計算機は 2.1GHz の SPARC64 プロセッサである。筆者らのフローでは従来の STA フローを使ってディレイ改善を行い、目標のディレイ改善ができた段階で SSTA を実行して周波数歩留まりを確認するため、SSTA の処理時間が実行フローのネックとはなっていない。

7. まとめ

筆者らは自社開発した SSTA ツールを実際のプロセッサ設計に適用して、プロセッサ設計者が周波数歩留まり向上のために改善すべき部分を定量的に把握できる仕組みを構築した。この仕組みは、現在開発中のプロセッサ設計において適用され、設計者に改善の指針として使われている。設計者は、STA によるパスディレイのチェックをクリアした後で SSTA を実行して、SSTA 用に新たに設定された周波数歩留り目標をクリアする。STA ではクリティカルな個々のパスを解析して、設計者はパスのディレイを改善する。設計者が見る SSTA の結果は周波数歩留り分布であり、個々のパスが SSTA ではどのように評価されているかを見ることはない。しかし今回、筆者らは STA と SSTA に対して、同じパスがクリティカルパスの順位の間でどのように変動しているかを比較し、その要因の分析を行った。このような分析を通して、STA と SSTA の特徴を明らかにし、設計フローの中で周波数歩留まり向上のための設計改善をより効率よく行えるようにツールやフローの改善を図って行く。

文 献

- [1] 三好章夫、横田俊彦，“IBM の 65nm のバラつき対策”，日経マイクロデバイス 2月号，pp. 53-59, 2007.
- [2] A. Nardi, E. Tuncer, et al, “Use of Statistical Timing Analysis on Real Designs”, *Design Automation, and Test in Europe*, 2007.
- [3] 本間克己、新田泉、澁谷利行，“パスベース遅延解析におけるチップ内・チップ間バラツキを考慮した遅延分布計算”，信学技報，vol. 106, no. 549, pp. 93-98, 2007年3月.
- [4] K. R. Heloue, F. N. Najm, “Statistical Timing Analysis with Two-sided Constraints”, *International Conference on Computer-Aided Design*, pp. 829 – 836, 2005.
- [5] N. Ito, et al, “A Physical Design Methodology for 1.3GHz SPARC64 Microprocessor”, *International Conference on Computer Design*, pp. 204-210, 2003.
- [6] A. Inoue, “SPARC64TM V/VI for Mission-Critical Servers,” presented at *Fall Processor Forum*, 2004.
- [7] 池田弘、小松裕成、その他，“統計的タイミング解析によるチップ内ばらつきの推定”，DA シンポジウム，2007.
- [8] 伊藤則之、小松裕成、その他，“プロセッサ設計における統計的タイミング解析の実適用”，DA シンポジウム，2007.