

LSI 電源設計検証のための順序回路の最大動作率解析

樋口 博之[†] 金澤 裕治[†] 盛山 修[†] 伊藤 則之^{††}

[†] (株) 富士通研究所 〒211-8588 川崎市中原区上小田中 4-1-1

^{††} 富士通 (株) 〒211-8588 川崎市中原区上小田中 4-1-1

E-mail: [†]{higuchi,kanazawa}@labs.fujitsu.com, ^{††}{moriyama.osamu,ito.noriyuki}@jp.fujitsu.com

あらまし 本稿では、LSI オンチップ電源設計検証のために、順序回路に対して動作率ができるだけ大きい入力パターンを効率的に生成する方法を提案する。本手法では ATPG ベースの同時動作ペア解析を導入することで、最大動作率解析問題を最大クリーク問題に帰着する。実験により大規模回路にも適用でき、質のよいパターンが得られることを示す。
キーワード オンチップ電源設計 電源ノイズ検証 最大動作率 順序回路 ATPG クリーク

Analysis of Maximum Switching Activities in Sequential Logic Circuits for Power Supply Integrity Validation

Hiroyuki HIGUCHI[†], Yuzi KANAZAWA[†], Osamu MORIYAMA[†], and Noriyuki ITO^{††}

[†] FUJITSU LABORATORIES LTD.

Kamikodanaka 4-1-1, Nakahara-ku, Kawasaki, Kanagawa, 211-8588

^{††} FUJITSU LIMITED

Kamikodanaka 4-1-1, Nakahara-ku, Kawasaki, Kanagawa, 211-8588

E-mail: [†]{higuchi,kanazawa}@labs.fujitsu.com, ^{††}{moriyama.osamu,ito.noriyuki}@jp.fujitsu.com

Abstract We present a method for calculating maximum switching activity in sequential circuits for power supply noise analysis. The approach is based on pair-wise simultaneous switching activity analysis using ATPG techniques for sequential circuits. The information obtained by simultaneous switching analysis between each gate pair is used to generate lower and upper bounds of maximum switching activity based on maximum clique formulation.

Key words on-chip power noise analysis, maximum switching activity, sequential circuit, ATPG, clique

1. はじめに

集積回路 (LSI) 技術の進歩とともに、マイクロプロセッサなど高性能 LSI 設計においてチップ内部の電源ノイズ対策および解析の重要性が増してきている [1]。0.13 μ m テクノロジでは、グラウンドや Vdd の電圧が 10% 変動するとゲート遅延は 30% 程度変動するといわれている [2]。また、低消費電力化のための電源電圧削減など最近の LSI 設計トレンドにより、電源ノイズマージンはますます減少し、回路誤動作の可能性が増している。誤動作のない、所望の性能を持つ LSI を設計するには、設計時に電源ノイズ解析を十分に行い、回路がどのような動作をしても電源ノイズが許容範囲に収まっていることを確認することが不可欠である。

LSI の電源ノイズ解析の第一の目的は電源網の信頼性検証であるが、第二の目的として、電源網設計のノイズ対策コスト最適化がある。電源ノイズ対策として、オンチップデカップリングキャパシタンスを挿入する方法 [3]~[7]、高動作率セルをチッ

プに拡散させホットスポットを削減する方法、電源・グラウンド網のトポロジ最適化を行う方法 [8], [9]、配線サイジングを行う方法 [10]、など様々な方法が提案されている。電源ノイズ削減に要するコストは年々増大しており、高性能 LSI では設計全体のコストに大きな影響を与えている。例えば、高性能マイクロプロセッサにおいては、チップ面積の 15-20% がデカップリングキャパシタンスで占められることもある [11]。過剰なオンチップデカップリングキャパシタンスはリーク電流増大にもつながるため、その最適化は重要な問題である。電源ノイズ対策として挿入されたデカップリングキャパシタンスが過剰であることが解析により分かれば、それを削減し、チップ面積の削減や機能の追加などを行うことができる。

電源網での電圧降下 (ΔV) は抵抗による電圧降下 (IR) とインダクタンスによる電圧降下 (LdI/dt) の和である ($\Delta V = IR + LdI/dt$)。電源網解析が難しい原因の一つは、電圧降下が回路の入力と状態に依存することである。この点に関するアプローチの違いにより電源網解析を大きく分類すると、入力パタ

ンを用いないスタティック解析と入力パターンを用いて解析を行うダイナミック解析の二つに分けられる。スタティック解析は高速であるが、パターンや状態に独立に解析を行うため、信号線間の論理的な依存関係を考慮せず、各信号線は論理的には任意の値、遷移をとりうるとして解析を行う。そのため、結果が悲観的、すなわち、最大動作率を必要以上に大きく見積もる傾向がある。さらに、近年インダクタンスによる電圧降下が無視できず、 IR と LdI/dt の両方を考慮する必要が出てきており、 IR のみの場合には最大電流のみを考えればよかったものが、 LdI/dt も含めるとそれに加えて最大電流変化も同時に考える必要があり、スタティックな解析ではさらに悲観性が大きくなる可能性が高い。一方、ダイナミック解析はパターンが与えられればスタティックよりも精度の高い解析ができるが、いかに質のよいパターンを用意するかが一番大きな問題である。論理シミュレーション用のパターンを用いる場合にはパターンと状態の空間に対する網羅率が極めて低いことが一般的であるため、電圧降下の点で良いパターン、すなわち、動作率の十分高いパターンが含まれていない場合が少なくない。動作率の高いパターンを直接求める方法についての研究も行われているが、大規模回路に対して実用的な時間で質の良いパターンを求めるまでには至っていない。

そこで本稿では、ダイナミックな電源ノイズ解析の入力パターンを用意することを目的として、大規模回路に対する効率的な最大動作率パターンの生成と動作率上界計算の方法を提案する。従来法で大規模回路に適用できない主な原因は、パターン生成で各パターンで動作率を計算しつつ、それが最大となるパターンを探索する必要があった。すなわち、動作する素子の探索と動作率最大パターンの探索という2つの探索を組み合わせる必要があった。本手法では2つの探索を切り離して行うために、素子ペアごとの同時動作可能性を調べる同時動作ペア解析を導入する。同時動作ペア解析の結果から得られる同時動作関係グラフの最大クリークを求める問題に帰着して最大動作率を求める。

2. 準備

2.1 用語の定義

本稿では、完全同期式順序回路を単に回路と呼ぶ。回路中の論理ゲートとフリップフロップ (FF) を素子と呼ぶ。素子の出力数は1とし、素子の出力値をその素子の値とも呼ぶ。素子の値が前時刻から現時刻で変化するとき、その素子は動作していると言う。ある時刻で回路中で動作している素子の数をその時刻での回路の動作素子数と呼び、動作素子数を全素子数で割ったものを回路の動作率と呼ぶ。回路が取りうる動作率の中で最大のものを最大動作率と呼ぶ。

ある時刻での回路の外部入力の値の並びを入力パターンと呼ぶ。ある時刻での回路のフリップフロップの値の並びを状態と呼ぶ。動作率は前時刻と現時刻の素子の値により決まるため、前時刻の入力パターンと状態、及び現時刻の入力パターンが決まれば、回路の動作率は一意に求められる。この、前時刻の入力パターンと状態、及び現時刻の入力パターンは回路を二時刻時間展開して得られる組み合わせ回路の入力パターンであるから、本稿ではこれを時間展開パターンと呼ぶ。回路の最大動作率を与える時間展開

パターンのことを本稿では最大動作率時間展開パターン、あるいは単に、最大動作率パターンと呼ぶ。本稿では回路の初期状態は仮定しない。すなわち全ての状態が到達可能であると仮定する。

本稿では、非巡回無向グラフを単にグラフと呼ぶ。グラフ G はノード集合 $V(G)$ と枝集合 $E(G)$ の組 $(V(G), E(G))$ で表される。グラフ G の部分グラフで、任意の2つのノードに枝が存在するグラフをクリークと呼ぶ。グラフ G でサイズが最大のクリークを最大クリークと呼ぶ。グラフ G のノードの集合で、任意の2つのノードに枝が存在しないノードの集合を独立集合と呼ぶ。グラフ G でサイズが最大の独立集合を最大独立集合と呼ぶ。グラフ G のノードに、「色」と呼ばれる整数 $1, 2, \dots, k$ を割り当て、隣り合う2つのノードにも同じ色がこないようにできることを k 彩色可能という。グラフ G が k 彩色可能であるような最小の整数 k のことを G の彩色数と呼ぶ。グラフ G の彩色数は最大クリークサイズの上界を与える。また、 G を独立集合で分割したときの分割数は G の彩色数の上界を与える。

2.2 最大動作率パターン生成の従来法

最大動作率解析の多くの従来法は組み合わせ回路を対象としている。組み合わせ回路の場合動作率は前時刻と現時刻の入力パターンの組、すなわち入力パターンペアにより定まる。まず、組み合わせ回路に対する最大動作率解析の従来法から説明する。

Chowdhury らはマクロに対して最大電流を与える入力パターンペアを生成する分枝限定法と近似手法を提案している [12]。近似手法では $m (> 1)$ 個の外部入力グループを選び、そのグループについてのみ全ての組み合わせを試す。近似手法を用いても対象回路規模は高々百ゲート程度である。

Devadas らは、最大消費電力解析において、最大動作率パターンを MAXSAT 問題に帰着して解く方法を提案している。動作率は unit delay のもとのハザードも考慮できるが、数百ゲート程度の回路の解析でも数時間を要している。

Krstic らは timed ATPG [13] を用いて最大電流を生成する入力パターンペアを生成する方法を提案している。この方法では、パターン独立な解析である iMax 法 [14] を行いて得られる電流の上界の waveform を出発点としてパターンペアを求める。上界の何パーセントという形で設定するカットオフ電流量 I_{cutoff} に対して、上界の waveform であり $\max(I_{iMax}(t))$ が I_{cutoff} を越える時刻 $t=T$ (固定遅延なので取りうる時刻の数は有限) について、timed ATPG を用いて $\max(I_{iMax}(T))$ の値が本当に起こりうるかを調べる。取りうる時刻 T が複数になる場合はそれぞれに対して入力パターンペアを生成する。timed ATPG は通常の、信号値だけを持つテストパターン生成 (ATPG) を拡張して、信号値とその信号値が到達する時刻の2つ組を持つような ATPG である。そのため、固定遅延のもとの遅延やグリッチを考慮できるが、大規模回路への適用は非現実的である。

Jiang らは、一つの入力パターンペアを生成するだけでなく、電源網解析の網羅性を考慮して、電圧降下がある程度以上のセル全てをチェックするような入力パターンペアの集合を生成する問題を考え出した。すなわち、電圧降下がある閾値を越えて電圧降下するセルの全てを少なくとも一度は大きく電圧降下させるような入力パターンペアの集合を求める問題をクリティカル

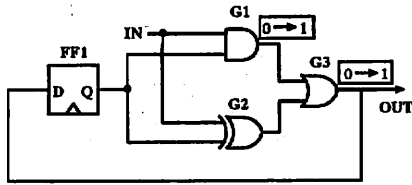


図 1 例

電圧降下問題と名付け、そのようなパターンペア集合を求める方法として GA を用いた方法を提案している [15].

次に順序回路を対象とした従来法について述べる。Wang らは、最大消費電力解析を順序回路を対象として効率的に行う方法を提案した [16]。本稿で対象とする最大動作率解析と [16] が対象としている問題の違いは、最大動作率解析では動作するゲート数をコストとするのに対して、[16] では動作する素子のファンアウト数の和をコストをとするという点のみである。順序回路は組み合わせ回路部分を 2 時刻時間展開し、展開した回路上で ATPG ベースのパターン探索を行う。大きなファンアウトの素子から順に欲張り法的 (greedy) に動作するゲートを探索していく単純なアルゴリズムのため、比較的大規模な回路に対しても適用可能である。

3. 同時動作ペア解析による最大動作率解析

3.1 例

本節では図 1 の順序回路を例にとり提案手法の基本的考え方を説明する。提案手法ではまず、各素子ペアについて、そのペアが同時動作しうるか、さらに同時動作するときの遷移値のペアは何か、を調べる。これを本稿では同時動作ペア解析と名付ける。図 1 で例として、素子ペア (G1, G3) の解析を考える。素子ペア (G1, G3) が同時に動作するということはそれらの信号値が変化するような時間展開パターン、すなわち前時刻の回路の状態パターンと前時刻と現時刻の入力パターンのペア、が存在するということである。素子ペアの信号遷移の組み合わせは (G1, G3) = (rise, rise), (rise, fall), (fall, rise), (fall, fall) の 4 通りが存在しうる。解析ではこれらそれぞれについて時間展開パターンが存在するか調べる。図 1 では (G1, G3) = (rise, rise) の場合のみ示している。図 1 ではゲート G1 と G3 の横の四角の枠で囲まれた部分が示すように、それぞれにまず立ち上がり遷移を割り当てる。四角の枠の中の矢印の左側の値が前時刻の信号値を示し、矢印の右側の値が現時刻の信号値を示している。この割り当てを満たすような時間展開パターンが存在するかどうかは図 1 の順序回路を 2 時刻時間展開した組み合わせ回路に対して ATPG ベースのパターン探索を行うことにより調べられる。その様子を図 2 に示す。この例では含意操作のみによりパターンが存在しないことが証明できている。一般的には、含意操作だけで証明できない場合は ATPG と同様バックトラックにより解空間を探索していくことになる。図 2 で破線の矢印は含意操作により矢印の根元の値から矢印の先の値が含意された、すなわち、一意的に得られたことを示す。この図から分かるように

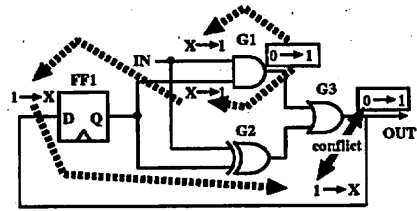


図 2 同時動作ペア解析

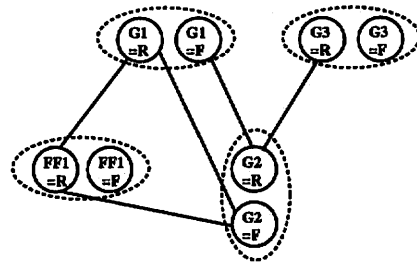


図 3 同時動作関係グラフ

ゲート G3 に新たに割り当てられた前時刻の値 1 が、前提条件として割り当てられていた値 0 と矛盾している。これはゲートペア (G1, G3) が同時に立ち上がり遷移を持つことがないことを示している。

各素子ペアに対する同時動作ペア解析の結果から同時動作関係グラフを作ることができる。同時動作関係グラフとは、そのグラフの各ノードが回路の素子とその遷移値の組に対応し、枝はそのノードペアに対応する回路素子ペアが指定遷移値をとる動作が同時に起こりうることを表したグラフである。例として用いている図 1 の回路の同時動作関係グラフは図 3 のようになる。各ノードには対応する素子とその素子の遷移値がラベルづけされている。“R” は立ち上がり遷移，“F” は立ち下がり遷移を表す。破線で囲まれたノードは同じ素子に対する (遷移値の異なる) ノードであることを示す。ある素子集合が同時動作するにはその集合内の任意の素子ペアが同時動作する必要がある。同時動作関係グラフではそのような集合はクリークとして現れるため、同時動作関係グラフから最大クリークを求めることにより同時動作する素子集合の上界集合が求められる。クリーク中の全ての素子ペアに対してそれぞれ同時動作する時間展開パターンが少なくとも一つ存在することは保証されているが、それらの素子ペア全てを同時動作させる共通のパターンが必ず存在するとは限らないので、この最大クリークのサイズは最大同時動作素子数の上界である。図 3 の場合には、最大クリークのノードは {FF1 = R, G1 = R, G2 = F} であり、得られる上界サイズは 3 と分かる。動作率は $(3/4) \times 100(\%) = 75(\%)$ である。

上界計算で得られた動作率の上界が実際に起こりうるかどうかは見つけ出した最大クリークの各ノードが表す遷移値の割り当てが実際に起こるかどうかが、すなわち、その割り当てを満たすような時間展開パターンが存在するかどうかを調べることにより

確かめられる。図3の最大クリークに対応する遷移値の割り当て $FF1 = R, G1 = R, G2 = F$ を実現する時間展開パターンを探ると、この例の場合 $IN(0) = 1, IN(1) = 1, FF(0) = 0$ というパターンが見つかる。ここで、前時刻を時刻0、現時刻を時刻1とし、例えば、 $IN(0) = 1$ は IN が時刻0で値1を取ることを表すものとする。以上の解析の結果、図1の例では最大動作率は厳密に75%であり、そのパターンは $IN(0) = 1, IN(1) = 1, FF(0) = 0$ であると分かる。

以下、同時動作ペア解析、上界計算、下界計算それぞれの処理について説明し、最後の節で全体の処理の流れを説明する。

3.2 同時動作ペア解析

同時動作ペア解析では与えられた順序回路を2時刻分時間展開した組み合わせ回路に対してテストパターン生成技術を適用する。これは前時刻と現時刻の間の順序回路としての状態遷移の情報を考慮することを意味する。前時刻の状態が到達可能かどうかは考慮しない。順序回路を2時刻分時間展開するとは順序回路の組み合わせ回路部分のコピーを時刻0の分と時刻1の分の2つ作り、時刻0の組み合わせ回路部分のFFの入力につながる信号線と時刻1の組み合わせ回路部分に対応するFFの出力につながる信号線を外部入力に含め、時刻1の組み合わせ回路部分のFFの入力を外部出力に含めた回路である。この時間展開した組み合わせ回路を用いることで時刻0から時刻2への遷移関係を考慮した解析を行うことができる。

同時動作ペア解析を用いた提案手法の大きな特徴は時間展開パターンの探索は同時動作ペア解析のときにほぼ限定して行われるということである。同時動作ペア解析のあとは、同時動作関係グラフのみを用いて最大同時動作素子集合の探索を行う。これにより時間展開パターンの探索と最大動作素子集合の探索を独立して行うことができる。

同時動作ペア解析は全素子ペアと全遷移値の組み合わせに対して行う。解析高速化のため3.5節で述べるように、乱数パターンやATPGパターンのシミュレーションをあらかじめ行い、同時動作することが容易に分かる素子ペアのドロップを行う。

3.3 最大クリークソルバによる最大動作率上界計算

3.2節で説明した同時動作ペア解析により、少なくとも一つの時間展開パターンに対して同時動作する全ての素子ペアとその遷移値を認識できる。与えられた回路の中で同時動作する素子の集合は、その素子の集合の要素の全てのペアが同時動作する必要がある。したがって、同時動作関係グラフの最大クリークのサイズは最大動作素子数の上界を表す。提案手法では、この考えに基づき同時動作関係グラフに対して最大クリークソルバを適用することにより最大動作率の上界計算を行う。

提案手法での上界計算が効率的かどうかは最大クリークソルバが効率的かどうか大きく依存する。我々は最大独立集合分割に基づく彩色数の下界を積極的に枝がりに利用するCoudertによる最大クリーク探索アルゴリズム[17]を重みづけグラフに拡張して実装している。

同時動作関係グラフが大きい場合には同じ素子に対する2つのノードをマージしたグラフを用いるか最大クリークサイズの上界を求めることで最大動作素子数の上界とする。最大クリーク

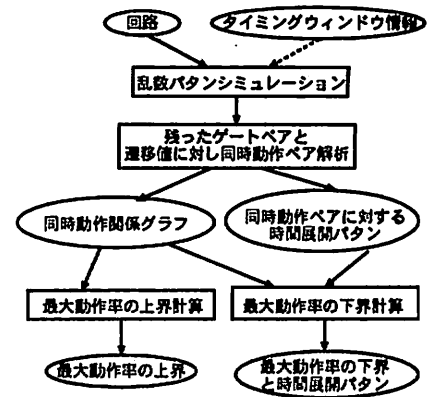


図4 全体の処理の流れ

クサイズの上界は同時動作関係グラフの彩色数の下界を求めることで得られる。彩色数の下界は、色のある割り当てが求まればそれが一つの下界となる。よい下界を得るため提案手法では欲張り法的 (greedy) に最大独立集合の分割を求めていくことで色の割り当てを求めている。

3.4 最大動作率下界計算とパターン生成

最大動作率の下界を与える時間展開パターンの生成ではできるだけ動作率が高いパターンを生成することが望ましい。そのため、まず前節の上界計算で厳密最大クリークが求まった場合にはそのクリークに対応する遷移値割り当てが実際に起こりうるか、すなわちそのような割り当てが起こる時間展開パターンが存在するかをATPGベースの探索により求める。そのようなパターンが存在すればそのパターンを最大動作パターンとして生成する。その場合前節で得られた上界が下界と一致した、すなわち厳密解であることが証明できたことになる。

最大クリークに対応する遷移値割り当てを満たすパターンがない場合、最大動作率の下界のパターンを求める。以下の二種類のパターンのうち動作率のより大きい方を用いる。

(1) 同時動作関係グラフから近似手法により、遷移値の割り当てを同時に満足できるできるだけ大きなクリークを求め、その遷移値割り当て全てを満足する時間展開パターンを生成する。

(2) 同時動作ペア解析で見つけられたパターンについて動作率を求めておき、その動作率が一番大きなパターンを生成する。

3.5 全体の処理の流れ

提案手法全体の処理の流れを図4に示す。入力回路記述で、必要に応じてタイミング解析で得られる各素子のタイミングウィンドウ情報も入力できる。タイミングウィンドウ情報を入力すれば各素子ペアの遷移タイミングも考慮した同時動作可能性が調べられ、対象素子ペア数削減につながる。乱数パターンシミュレーションは容易に同時動作可能と分かる素子ペアのドロップを行い、ドロップできなかった素子ペアに対してのみ同時動作ペア解析を行う。

4. 実験結果

本稿で述べた同時動作ペア解析に基づく最大動作率解析アル

ゴリズムを C++ 言語で実装し、ISCAS89 順序回路ベンチマーク [18] に対して最大動作率の上界と下界および下界に対応する時間展開パターンを生成する実験を行った。本実験を行った計算機は SPARC64 V(1100MHz, 主メモリ 16GB), OS は Solaris 9, 使用コンパイラは gcc version 2.95.3, コンパイラオプションは "-O2" である。実験では表 4 にあるタイミング解析のタイミングウィンドウ情報は用いていない。

実験結果を図 1 に示す。表 1 の各項目は次の通りである。欄「 N_{PI} 」は外部入力数を示す。欄「 N_{FF} 」は FF 数を示す。欄「 N_{gate} 」はゲート数を示す。欄「Max.Activity」は解析の結果となる動作率を示す。欄「LB」は最大動作素子数の下界を示す。括弧内は最大動作率の下界を百分率で示したものである。欄「UB」は最大動作素子数の上界を示す。括弧内は最大動作率の上界を百分率で示したものである。欄「RSim」は乱数パターンを用いたシミュレーションで一番大きな動作素子数を示す。括弧内はその動作率を百分率で示したものである。欄「RSim#Pat」は乱数パターン数を示す。欄「CPU time」は最大動作率解析に要した CPU 時間を示す。欄「SSP」は同時動作ベア解析に要した CPU 時間を示す。欄「UB」は上界計算に要した CPU 時間を示す。欄「LB」は下界計算に要した CPU 時間を示す。「h」は時間、「m」は分、「s」は秒をそれぞれ示す。最下行の「average」は各欄の数の平均(総和を回路数で割ったもの)を示す。

表 1 より、提案手法は全ての ISCAS ベンチマークに対してパターン生成ができていることが分かる。十分多くの乱数によるシミュレーションで得られる動作率に比べ、提案手法の下界計算で得られる動作率は 8.2% 大きく、電源ノイズ解析に有効なパターンが得られたと考える。最大動作率の下界は 54.8%, 上界は 74.6% で、その幅は 19.8% であった。上界計算はゲート数が 2 万程度の回路は 15 時間で計算が終了しなかった。上界計算については同時動作関係グラフの簡約化や最大クリークソルバの近似ヒューリスティックの改善などにより性能改善をさらに行う必要がある。

5. おわりに

本稿では、LSI オンチップ電源設計検証のために、順序回路に対して最大動作率の上界と動作率のできるだけ大きなパターンを効率的に生成する方法を提案し、ISCAS89 ベンチマークを用いてその評価を行った。実験結果から本手法は大規模回路でも適用可能で、得られるパターンは乱数シミュレーションによるものより動作率が 8.2% 大きいことが分かった。今後、提案手法で得られたパターンを用いて電源ノイズ解析を行う予定である。

文 献

- [1] 伊藤則之, 吉竹昭博, 小松裕成, 芳野弘宜, 鍋谷研一, 天野靖雄, 畑村佳保, 河村吉嗣, and C. K. Cheng. マイクロプロセッサ設計における電源網設計・解析とその応用. In *DA シンポジウム 2006*, July 2006.
- [2] S. Pant, D. Blaauw, V. Zolotov, S. Sundareswaran, and R. Panda. Vectorless analysis of supply noise induced delay variation. In *Proceedings of the International Conference on Computer Aided Design*, pages 184–191, November 2003.
- [3] H.-M. Chen, L.-D. Huang, I.-M. Liu, M. Lai, and D. F. Wong. Flooplanning with power supply noise avoidance. In

- Proceedings of Asia and South Pacific Design Automation Conference*, pages 427–430, January 2003.
- [4] S. Zhao, K. Roy, and C. K. Koh. Decoupling capacitance allocation and its application to power-supply noise-aware flooplanning. *IEEE Transactions on Computers*, pages 81–92, January 2002.
- [5] M. Graziano, G. Masera, G. Piccinini, and M. Zamboni. Automated power supply noise reduction via optimized distributed capacitors insertion. In *Proceedings of Southwest Symposium on Mixed-Signal Design*, pages 167–172, February 2001.
- [6] H. Hu, S. S. Sapatnekar, and S. R. Nassif. Optimal decoupling capacitor sizing and placement for standard cell layout designs. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, pages 428–436, April 2003.
- [7] C. Y. Yeh and M. Marek-Sadowska. Timing-aware power noise reduction in layout. In *Proceedings of International Conference on Computer Aided Design*, pages 627–634, 2005.
- [8] G. Chen and S. Sapatnekar. Partition-driven standard cell thermal placement. In *Proceedings of the International Symposium on Physical Design*, pages 75–80, 2003.
- [9] B. Obermeier and F. M. Johannes. Temperature-aware global placement. In *Proceedings of the Asia South Pacific Design Automation Conference*, pages 143–148, January 2004.
- [10] R. Dutta and M. Marek-Sadowska. Automatic sizing of power/ground (P/G) networks in VLSI. In *Proceedings of the 26th Design Automation Conference*, pages 783–786, June 1989.
- [11] P. E. Gronowski, W. J. Bowhill, R. P. Preston, M. K. Gowan, and R. L. Allmon. High-performance microprocessor design. *IEEE Journal of Solid-State Circuits*, 33(5):676–686, May 1998.
- [12] S. Chowdhury and J. S. Barkatullah. Estimation of maximum currents in MOS IC logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, 9(6):642–654, June 1990.
- [13] S. Devadas, K. Keutzer, and S. Malik. Computation of floating mode delay in combinational circuits: Theory and algorithms. *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, 12(12):1913–1923, December 1993.
- [14] H. Kriplani, F. N. Najm, and I. N. Hajj. Pattern independent maximum current estimation in power and ground buses of CMOS VLSI circuits: Algorithms, signal correlations, and their resolution. *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, 14(8):998–1012, August 1995.
- [15] Y. M. Jiang, T. K. Young, and K. T. Cheng. VIP – an input pattern generator for identifying critical voltage drop for deep sub-micron designs. In *Proceedings of the International Symposium on Low Power Electronics and Design*, pages 156–161, 1999.
- [16] C. Y. Wang, K. Roy, and T.-L. Chou. Maximum power estimation for sequential circuits using a test generation based technique. In *Proceedings of IEEE Custom Integrated Circuits Conference*, pages 229–232, May 1996.
- [17] O. Coudert. Exact coloring of real-life graphs is easy. In *Proceedings of the Design Automation Conference*, pages 121–126, June 1997.
- [18] D. Bryan F. Brglez and K. Kosminski. Combinational profiles of sequential benchmark circuits. In *Proceedings of the International Symposium on Circuits and Systems*, May 1989.

表 1 実験結果

circuit	N_{PI}	N_{PF}	N_{gate}	Max. Activity			RSim	#Pat.	CPU time		
				LB	UB	RSim			SSP	LB	UB
s208.1	10	8	104	70 (63%)	70 (63%)	68 (60%)	2,048	0s	0s	0s	
s208	11	8	96	61 (59%)	61 (59%)	61 (59%)	2,368	0s	0s	0s	
s298	3	14	119	91 (68%)	105 (79%)	83 (62%)	1,664	0s	0s	0s	
s344	9	15	160	125 (71%)	144 (82%)	107 (61%)	1,024	0s	1s	0s	
s349	9	15	161	126 (72%)	145 (82%)	107 (61%)	1,024	0s	1s	0s	
s382	3	21	158	118 (66%)	151 (82%)	108 (60%)	1,504	0s	1s	0s	
s386	7	6	159	90 (55%)	115 (70%)	85 (52%)	2,784	0s	1s	0s	
s400	3	21	164	122 (66%)	152 (82%)	109 (59%)	2,240	0s	1s	0s	
s420.1	18	16	218	139 (60%)	141 (60%)	101 (43%)	6,272	2s	2s	0s	
s420	19	16	196	116 (55%)	130 (61%)	96 (45%)	4,736	1s	2s	0s	
s444	3	21	181	120 (60%)	159 (79%)	118 (58%)	2,720	0s	5s	0s	
s510	19	6	211	92 (42%)	114 (53%)	90 (42%)	1,824	1s	3s	1s	
s526	3	21	193	131 (61%)	115 (54%)	116 (54%)	3,488	0s	2s	1s	
s526n	3	21	194	134 (62%)	154 (72%)	116 (54%)	3,488	0s	2s	0s	
s641	35	19	379	302 (76%)	362 (91%)	246 (62%)	7,008	0s	6s	0s	
s713	35	19	393	308 (76%)	361 (88%)	246 (60%)	7,008	0s	5s	0s	
s820	18	5	289	145 (49%)	203 (69%)	143 (49%)	4,608	1s	9s	0s	
s832	18	5	287	140 (48%)	200 (69%)	138 (47%)	4,608	1s	29s	0s	
s838.1	34	32	446	275 (58%)	281 (59%)	117 (25%)	8,256	18s	23s	41s	
s838	35	32	390	198 (47%)	244 (58%)	160 (38%)	23,712	9s	29s	0s	
s953	16	29	395	156 (37%)	236 (56%)	114 (34%)	13,440	9s	1h00m00s	32s	
s1196	14	18	529	263 (48%)	382 (70%)	228 (42%)	12,480	2s	1h00m00s	0s	
s1238	14	18	508	238 (45%)	361 (69%)	207 (39%)	17,152	2s	1h00m00s	0s	
s1423	17	74	657	406 (56%)	638 (87%)	404 (55%)	5,568	1s	1h00m00s	3m11s	
s1488	8	6	653	321 (49%)	463 (70%)	310 (47%)	6,016	8s	56s	22s	
s1494	8	6	647	311 (48%)	453 (69%)	301 (46%)	6,016	8s	59s	32s	
s5378	35	179	2,779	1,764 (60%)	2,557 (86%)	1,398 (47%)	50,912	33s	1h00m00s	15m38s	
s9234.1	36	211	5,597	3,212 (55%)	5,460 (94%)	2,509 (43%)	91,456	5m23s	1h00m00s	2h40m21s	
s9234	19	228	5,597	2,657 (46%)	5,018 (94%)	2,405 (41%)	127,392	5m53s	1h00m00s	2h30m26s	
s13207.1	62	638	7,951	4,277 (50%)	7,099 (83%)	3,108 (36%)	121,696	32m02s	1h00m00s	5h48m10s	
s13207	31	669	7,951	4,042 (47%)	6,677 (78%)	2,929 (34%)	124,288	26m36s	1h00m00s	6h14m09s	
s15850.1	77	534	9,772	4,220 (41%)	9,273 (90%)	4,228 (41%)	176,864	16m59s	1h00m00s	14h08m29s	
s15850	14	597	9,772	3,952 (38%)	7,855 (76%)	3,441 (33%)	115,552	33m47s	1h00m00s	12h24m50s	
s35932	35	1,728	16,065	10,418 (59%)	-	9,165 (52%)	224	8m42s	1h00m00s	-	
s38417	28	1,636	22,179	8,381 (35%)	-	8,113 (34%)	208,384	2h45m06s	1h00m00s	-	
s38584.1	38	1,426	19,253	10,174 (49%)	-	9,138 (44%)	341,216	9h14m17s	1h00m00s	-	
s38584	12	1,452	19,253	8,819 (43%)	-	8,963 (43%)	372,256	7h11m13s	1h00m00s	-	
s499	1	22	152	69 (40%)	74 (43%)	66 (38%)	320	4s	1s	1s	
s635	2	32	286	224 (70%)	228 (72%)	106 (33%)	4,416	3s	3s	5s	
s938	35	32	390	275 (58%)	281 (59%)	117 (25%)	8,256	18s	22s	13s	
s967	16	29	394	177 (42%)	259 (61%)	164 (39%)	100,336	6s	1h00m00s	5s	
s991	65	19	519	387 (72%)	503 (94%)	326 (61%)	1,728	0s	1h00m00s	19s	
s1269	18	37	569	367 (61%)	552 (91%)	338 (56%)	1,792	0s	1h00m00s	11s	
s1512	29	57	780	467 (56%)	680 (81%)	382 (46%)	9,952	1s	1h00m00s	48s	
s3271	26	116	1,572	1,059 (63%)	1,422 (84%)	993 (59%)	11,360	2s	1h00m00s	8m05s	
prolog	36	136	1,601	861 (50%)	1,334 (77%)	823 (47%)	15,008	20s	1h00m00s	16m25s	
s3330	40	132	1,789	1,111 (58%)	1,602 (83%)	975 (51%)	20,928	3s	1h00m00s	22m32s	
s3384	43	183	1,685	1,159 (62%)	1,714 (92%)	1,064 (57%)	5,632	1s	1h00m00s	15m43s	
s4863	49	104	2,342	1,555 (64%)	2,348 (96%)	1,449 (59%)	2,976	1s	1h00m00s	44m57s	
s6669	83	239	3,319	1,800 (54%)	2,958 (89%)	1,658 (50%)	5,312	2s	1h00m00s	1h50m57s	
average				(54.8%)	(74.6%)	(46.6%)					