

高速トランジスタ配置を用いたセル内寄生見積もり手法

吉田 浩章[†] Kaushik De^{††} Vamsi Boppana^{††} 池田 誠^{†††} 浅田 邦博^{†††}

[†] 東京大学 大学院 工学系研究科 電子工学専攻

^{†††} 東京大学 大規模集積システム設計教育研究センター (VDEC)

〒 113-8656 東京都文京区本郷 7-3-1

^{††} Zenasis Technologies, Inc.

1671 Dell Avenue, Suite #100, Campbell, CA 95008, USA

E-mail: [†]{hiroaki,iked,asada}@silicon.u-tokyo.ac.jp, ^{††}{kaushik,vamsi}@zenasis.com

あらまし 最近我々は回路のトポロジ解析に基づいたセル内の寄生見積もり手法を提案した [1]. この手法は簡単なスタンダードセルに対しては有効であることが示されているが, 特に複雑なセルに対してはその精度は十分ではない. また簡単なセルに対しても, 高い精度を得るためには慎重な校正が必要となっている. これらの問題を解決するため, 本論文では非常に高速なトランジスタ配置手法を用いることによって, 校正を行うことなく, 複雑なセルに対しても正確な寄生見積もりを可能とする手法を提案する. 最後に提案手法の例題に対する計算機実験の結果を示し, 本手法の妥当性を示す.

キーワード スタンダードセル, セル内寄生見積もり, トランジスタ配置, トランジスタレベル最適化

Accurate Pre-layout Estimation of Intra-cell Parasitics Using Fast Transistor-level Placement

Hiroaki YOSHIDA[†], Kaushik DE^{††}, Vamsi BOPPANA^{††}, Makoto IKEDA^{†††}, and Kunihiro ASADA^{†††}

[†] Department of Electronic Engineering, University of Tokyo

^{†††} VLSI Design and Education Center(VDEC), University of Tokyo

7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan

^{††} Zenasis Technologies, Inc.

1671 Dell Avenue, Suite #100, Campbell, CA 95008, USA

E-mail: [†]{hiroaki,iked,asada}@silicon.u-tokyo.ac.jp, ^{††}{kaushik,vamsi}@zenasis.com

Abstract Recently we proposed a pre-layout estimation method of intra-cell parasitics based on topology analysis [1]. Although the paper showed that the parasitics inside simple cells could be estimated very accurately, it performs a poor estimation on complex cells. Additionally, even for such simple cells, it requires a deliberate calibration to obtain accurate estimates. To overcome these drawbacks, this paper proposes a new estimation method based on a fast transistor-level placement algorithm. Our experiment on an industrial standard cell library demonstrates the validity of the new method.

Key words Standark cells, intra-cell parasitic estimation, transistor placement, transistor-level optimization

1. はじめに

トランジスタレベル最適化は現在まで様々な手法が提案されており, ゲートレベル最適化に比べ非常に高い性能を得ることが可能であることが知られている. 代表的なものとしては, トランジスタサイジング [2][3] やマクロセルに基づいた手法 [4] などが挙げられる. また, 最近になってスタンダードセルに基づいたフローを対象にした

トランジスタレベル最適化手法が提案されている [5]-[7]. これはセルレイアウト生成手法の最近の著しい進歩に因るところが大きい [8][9]. これらの手法は徐々に高性能セルライブラリ設計に採用され始めている [10].

半導体プロセスの微細化に伴って VLSI の大規模化が進むにつれ, 設計の早い段階においてレイアウトや物理的な影響を考慮することが必須になってきている [11]. ゲー

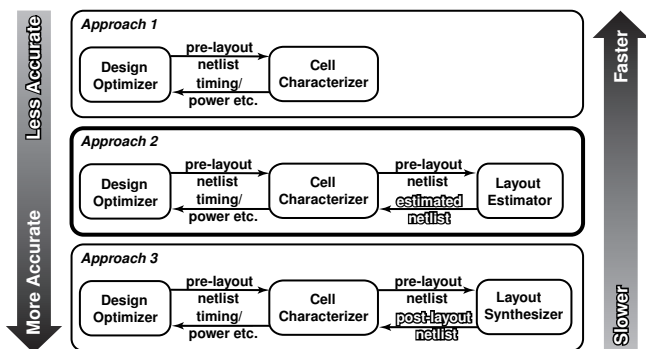


図1 トランジスタ最適化フローの比較

トレベル最適化は伝統的にこれらの影響を無視してきたが、これによって性能見積りに大きな誤差が生じてしまい、結果として最適化が収束しないという問題が発生した。この問題を解決するため、ゲートレベルの段階でレイアウトの影響を見積もり、それに基づいて最適化を行う手法が提案された。この手法に基づいた商用ツールはすでにいくつも存在しており、現在の超大規模設計には欠かせないものとなっている。

同様に寄生成分の影響はスタンダードセル内においても無視できなくなってきた。一方で、従来のトランジスタレベル最適化手法 [2] [12] はレイアウトによる寄生成分の影響を無視してきた。また、ほとんどの手法はトランジスタのモデルとして RC モデルを使用している。現在のサブミクロンレベルのトランジスタの特性は非常に複雑になっており、RC モデルを用いることはあまりに現実的でない。このような理由で、トランジスタレベル最適化においても、BSIM3/4 [13] に代表される正確なモデルを使用し、寄生容量を考慮しながらシミュレーションを行うことが必須であると考えられる。

シミュレーションに基づいたトランジスタレベル最適化のフローとして考えられる最も簡単なものは、レイアウト前のトランジスタ回路をシミュレーションし、その結果に基づいて最適化を行う手法である (図 1 Approach 1)。このフローはレイアウト生成を行わないため高速ではあるものの、レイアウトによる影響を完全に無視するために得られるタイミングや消費電力は非常に不正確なものとなる。一方で、レイアウトを生成しその抽出結果をシミュレーションするフローが考えられる (図 1 Approach 3)。このフローによって得られるセル情報は正確ではあるが、レイアウト生成はシミュレーションや抽出に比べ非常に遅いため、結果的に最適化に非常に多くの時間が必要となってしまふ。我々は図 1 の Approach 2 に示すようなフローを提案した [1]。これは、まずセル内の寄生容量の見積もりを高速に行い、次にこれを考慮してシミュレーションを行うフローとなっている。図 2 にレイアウト前と見積もり後のトランジスタ回路のモデルを示す。見積もり後の回路では、セルの高さに合わせてトランジス

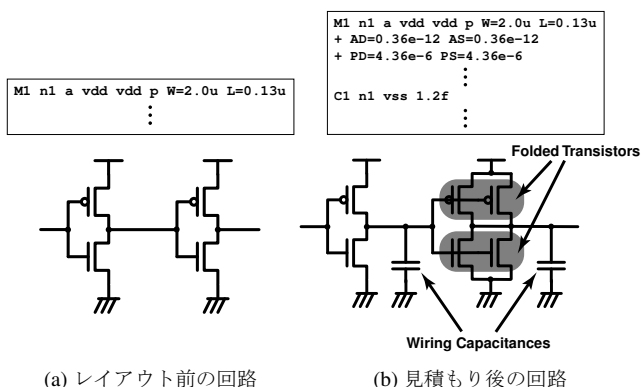


図2 トランジスタ回路モデル

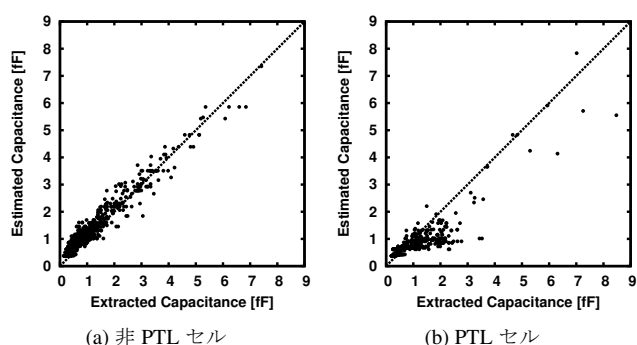


図3 抽出された配線容量と従来手法による見積もりの比較

タは折り畳まれ、配線容量と拡散の周囲長・面積が見積もられている。[1] では、これらの要素の見積もり手法として回路のトポロジ解析に基づいた手法を提案している。

このトポロジ解析に基づいたセル内寄生容量見積もり手法は簡単なスタンダードセルに対しては有効であることが示されているが、特に複雑なセルに対してはその精度は十分ではない。あるスタンダードセルライブラリの複数のセルの配線について、抽出された容量と従来手法による見積もりの比較を図 3 に示す。(a) は NAND や AOI ゲートに代表される PTL(Pass Transistor Logic) を用いないセル内の配線に、(b) は XOR やマルチプレクサなどといった PTL を用いているセル内の配線に対応している。PTL セルは非 PTL セルに比べ金属配線数が圧倒的に多く、また複雑である。図が示すように、非 PTL セルについては非常に正確な見積もりを実現しているが、PTL セルについてはその精度は低い。非 PTL セルが多くても 2 段程度の論理段からなるのに対して、PTL セルは複数の論理段からなっている。また中間段を接続する配線の容量の誤差は、入出力の配線に比べてシミュレーション結果に大きく影響する。このような理由で、複雑なセル内の配線の見積もり精度を向上させることは非常に重要である。またこの手法はプロセス非依存のトポロジ解析に基づいているため、簡単なセルに対しても高い精度を得るためには慎重な校正が必要となっている。

このような背景から、本論文では非常に高速なトランジスタ配置手法を用いて配置の見積もりを行うことによって、校正を行うことなく複雑なセルに対しても正確な寄

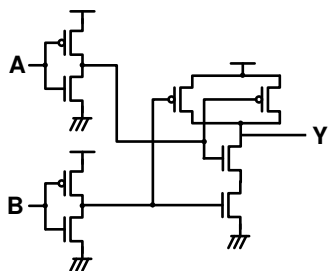


図4 2入力ORゲートのトランジスタ回路

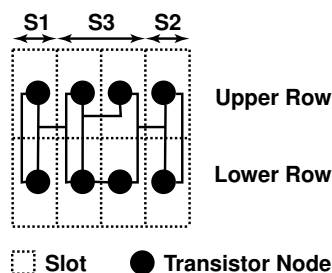


図6 配置モデル

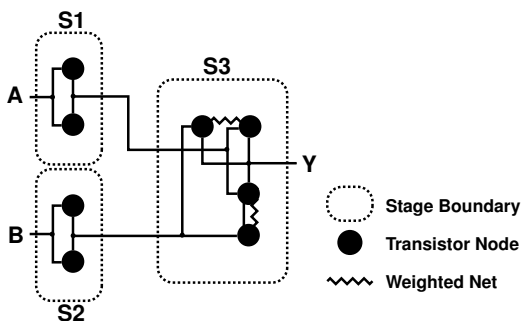


図5 2入力ORゲートの階層的ネットワーク

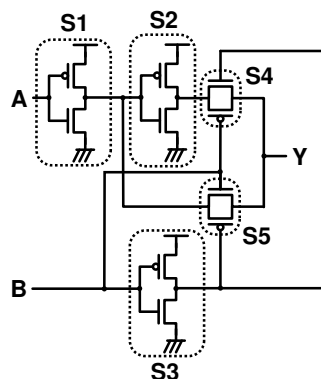


図7 2入力XORゲートのステージ分割例

生見積もりを可能とする手法を提案する。また提案手法の例題に対する計算機実験の結果を示し、本手法の妥当性を示す。

2. 高速トランジスタ配置手法

2.1 手法の概略

第1節で説明したように、提案手法では見積もりを高速に行うことが最も重要となる。具体的には次のステップであるシミュレーションの実行時間に比べて十分速いことが求められる。この高速性を実現するため、提案手法は二つの手法に基づいている。

- グラフ2分割手法に基づいた配置手法
- 階層的配置手法

グラフ2分割手法に基づいた配置手法[14]は、最も良く知られた配置手法の一つであり、効率の良いグラフ2分割手法を用いることで非常に高速な配置が可能である。階層的配置手法は、まずセルのトランジスタ回路をステージと呼ばれる概念によって分割し、ステージのネットワークとステージ内のトランジスタのネットワークという2つの階層によって表現する。ステージについては2.2節で詳しく説明する。次にステージの配置を決定し、その配置に基づいてステージ内のトランジスタの配置を決定する。図4の2入力ORゲートに対応する階層的ネットワークの例を図5に示す。この階層的配置手法には、解空間が制限されるという欠点が存在する一方、以下のような複数の有利な点が存在する。

妥当な配置への誘導: 一般的にステージ内のトランジスタは拡散部分を最大限利用して配線が行われるようにレイアウトされるため、お互い近くに配置される。

計算量の削減: 各々の階層における配置問題の大きさはフラットに配置を行う場合に比べ小さいため、全体の計算量の削減が可能である。

インクリメンタルな更新: 一般的にトランジスタレベル最適化では、ある最適化ステップにおいてごく一部のトランジスタしか変更しない。ステージ階層のトポロジが同じ場合には、変更のあったステージ内の配置の更新のみを行えば十分である。

本手法で使用する配置モデルを図6に示す。配置空間は上段と下段の2段に、また左右方向の複数列に分割されており、分割されたそれぞれの空間をスロットと呼ぶ。1つのスロットには多くても1つのトランジスタが配置可能であり、P型トランジスタは上段に、N型トランジスタは下段に配置される。また、この上下段からなるスロットの列は左右方向に分割され、それぞれがあるステージに対応する。あるステージ内のトランジスタはそのステージのスロット列にのみ配置可能である。図では、一番左の上下段がステージS1に、真ん中の2列がステージS3に、一番右がステージS2に対応している。

回路の階層化、ステージおよびステージ内トランジスタの配置手法については以下の節でそれぞれ詳しく説明する。

2.2 階層的ネットワークの構築

見積もりの対象となるセルのトランジスタ回路が与えられると、セルの高さに合うようにトランジスタの折り畳みを行う。本手法では[1]と同様の手法を用いている。具体的には配置後の全体の幅が最小になるようにトラン

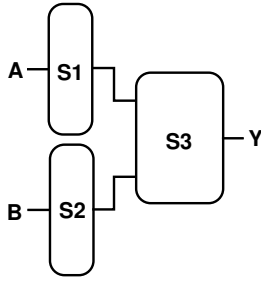


図8 ステージ階層ネットワーク

1. 初期分割 A, B を生成する
2. A と B から最もゲインの高い節を1つずつ選ぶ
3. 移動した後に (2) 式の条件を満たさないものは捨てる
4. 残っている節のうち最もゲインの高いものを選ぶ
5. その節を反対側へ移し, ロックする
6. 2-6 を繰り返す
7. すべての節のロックを解除する
8. カット数が削減された場合, 2-8 を繰り返す

図9 F-M アルゴリズムの手続きの概略

ジスタの折り畳みを行う。

次にこの回路をステージに分割することによって階層的ネットワークを構築する。分割については様々な手法が提案されている [17][18]。本論文では、電源に対応するネットを除いたネットワークにおいて、ドレインまたはソースによって接続されたトランジスタの最大集合をステージと定義している。ただし、例外的にトランスミッションゲートは一つの独立したステージとして扱う。図7に2入力 XOR ゲートの階層的ネットワークの例を示す。

最後に、同型のトランジスタの2つ以上のドレインまたはソースを接続しているネットについて、それらのドレインやソースを接続するネットを新たに追加する。図5中の波線で示されているネットがその例である。このように同型トランジスタのドレインまたはソースを接続するネットに重みを与えることによって、対応するトランジスタが近くに配置されるようになり、拡散による接続の最大化を実現している。

2.3 ステージ配置

ステージおよびステージ間の接続は、図8に示すようなネットワークによって表現できる。このネットワークでは、節がそれぞれのステージに対応している。ステージ配置問題は、これらの節の集合の一次元配置を決定する問題に帰着できる。この問題は既存のグラフ2分割手法に基づいた配置手法によって解くことが可能である。

本手法ではグラフ2分割アルゴリズムとして、F-M アルゴリズムと呼ばれる Fiduccia と Matheyses によるヒューリスティック [15] を使用している。以下にこの手法を簡単に説明する。あるグラフの節の集合の分割 A と B において、あるネットが A の節と B の節を接続しているとき、

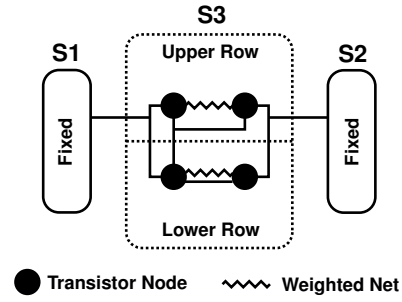


図10 トランジスタ階層ネットワーク

1. 初期分割 A_u, B_u, A_l, B_l を生成する
2. A_u, B_u, A_l, B_l から最もゲインの高い節を1つずつ選ぶ
3. 移動した後に (5) 式の条件を満たさないものは捨てる
4. 残っている節のうち最もゲインの高いものを選ぶ
5. その節を反対側へ移し, ロックする
6. 2-6 を繰り返す
7. すべての節のロックを解除する
8. カット数が削減された場合, 2-8 を繰り返す

図11 2段レイアウトスタイルに対応した F-M アルゴリズムの手続きの概略

そのネットはカットと呼ばれる。カット数はその分割におけるカットの合計数と定義される。このアルゴリズムはカット数を最小にする分割を決定することを目的としている。ある節のゲインはその節を反対側の集合に移したときのカット数の変化量として定義される。また分割のバランス R は次の式によって与えられる。

$$R = \frac{|A| - |B|}{2} \quad (1)$$

2つの分割間のバランスを保つため、 R は常に下の条件を満たす必要がある。

$$-1 \leq R \leq 1 \quad (2)$$

このアルゴリズムの手続きを図9に示す。このアルゴリズムの計算量はグラフ中の端点の数を n として $O(n)$ となっており、非常に高速である。

2.4 ステージ内トランジスタ配置

ステージ内トランジスタ配置では、図10に示すようなネットワークを使用する。このネットワークでは、節がそれぞれのトランジスタに対応しており、また外部ステージも節として表現される。ここで外部ステージとは現在注目しているステージ以外のステージのことを指している。トランジスタに対応する節はそのトランジスタの種類 (P型またはN型) に応じて上段または下段に属している。また、外部ステージに対応する節はステージ配置の結果に応じてトランジスタ部分の左側または右側に固定される。これによって、ステージ配置を考慮したトランジスタ配置が可能になっている。

この配置問題は完全な一次元配置問題ではないため、

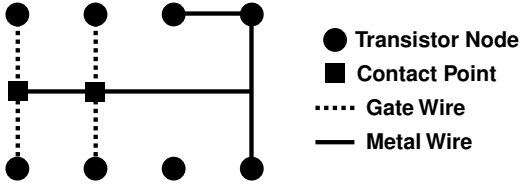


図 12 配線モデル

F-M アルゴリズムをそのまま使用することはできない。したがって、本手法では F-M アルゴリズムを拡張し、2 段レイアウトスタイルに対応したアルゴリズムを開発した。基本的な手続きは F-M アルゴリズムと同様であるが、分割された 2 集合間だけではなく上下段のバランスを考慮するためにバランスを再定義している。分割された 2 つの節の集合を $A_u(A_l)$ と $B_u(B_l)$ とする。 A_u と B_u は上段のトランジスタの集合、 A_l と B_l は下段のトランジスタの集合である。このとき分割のバランス R は次の式によって与えられる。

$$R = \frac{|A_d| - |B_d|}{2} \quad (3)$$

ここで A_d , B_d , A_n , B_n は次のように定義される。

$$\begin{aligned} A_d = A_u, B_d = B_u, A_n = A_l, B_n = B_l \\ (|A_u| + |B_u| \geq |A_l| + |B_l| \text{ の場合}) \\ A_d = A_l, B_d = B_l, A_n = A_u, B_n = B_u \\ (|A_u| + |B_u| < |A_l| + |B_l| \text{ の場合}) \end{aligned} \quad (4)$$

このとき、2 つの分割間のバランスは以下の条件で保たれる。

$$-1 \leq R \leq 1 \text{ かつ } A_d \geq A_n \text{ かつ } B_d \geq B_n \quad (5)$$

このアルゴリズムの手続きを図 11 に示す。このアルゴリズムの計算量も F-M アルゴリズムと同様に、グラフ中の端点の数を n として $O(n)$ となる。

3. 配置情報を用いた配線容量見積もり

本論文ではセル内寄生成分の中でも特に問題としている配線容量の見積もりのみを扱う。また対地容量と線間容量の合計を対地容量とみなし、対地容量のみを見積もる。本節では、第 2 節の手法によって決定された配置に基づいて配線容量を見積もる手法を説明する。

本手法で使用される配線モデルを図 12 に示す。配線はスタイナ木によってモデルされており、ゲート配線と金属配線の部分に分かれている。コンタクト点はゲートに対応する端子を通る垂直線と上下段の中間線の交点と定義される。ゲート配線はゲートに対応する端子とその直下(直上)にあるコンタクト点を結ぶ線分である。金属配線はドレインまたはソースに対応する端子とコンタクト点を結ぶ最小スタイナ木である。本手法では最小スタイナ木の構築に [16] に基づいたアルゴリズムを使用している。

表 1 実験に使用したライブラリセルの概要

セルの種類	総セル数	総配線数
非 PTL セル	241	1166
PTL セル	54	520
全体	295	1686

配線容量はゲート配線と金属配線からなるスタイナ木の合計長にテクノロジー依存のパラメータを掛け合わせることで求まる。

4. 実験結果

本論文の提案手法の実装を行い、例題を用いた計算機実験を行った。今回は $0.13\mu\text{m}$ プロセスによる商用スタンダードセルライブラリを例題として使用した。表 1 に使用したライブラリセルの概要を示す。

図 13 に配線容量の抽出値と従来手法による見積もりとの比較をそれぞれ示す。図 14 は同様の比較を提案手法を用いて行ったものである。配線容量の抽出値は、セルレイアウトから抽出された集中容量ネットワークを用いて、対応する配線の対地容量と線間容量の合計から求めたものである。従来手法のための学習に使用するセルとしては総セルの 10% を選んだ。具体的には非 PTL セルが 24 個、PTL セルが 5 個である。提案手法はすべての配線の見積もりを 1 秒以内に終了した。

表 2 に配線容量の抽出値と見積もり値の誤差平均と標準偏差の比較を示す。表において誤差平均 \bar{E} と標準偏差 S は以下の式で計算したものである。

$$E(i) = \left| \frac{C_{est}(i) - C_{ext}(i)}{C_{ext}(i)} \right| \quad (6)$$

$$\bar{E} = \frac{1}{n} \sum_{i=1}^n E(i) \quad (7)$$

$$S = \sqrt{\frac{1}{n} \sum_{i=1}^n (E(i) - \bar{E})^2} \quad (8)$$

ここで $C_{ext}(i)$ は i 番目の配線の容量の抽出値、 $C_{est}(i)$ は i 番目の配線の容量の見積もり値である。表において、改善度は従来手法と提案手法の誤差平均または標準偏差の差である。

5. まとめと今後の課題

本論文では非常に高速なトランジスタ配置手法を用いることによって、校正を行うことなく、複雑なセルに対しても正確な寄生見積もりを可能とする手法を提案した。また例題を用いた計算機実験では、従来手法に比べて誤差平均、標準偏差ともに約 10% の改善が実現されたことを示し、本手法の妥当性を示した。

しかしながら、今回の実験結果が示すように、特に複雑なセルの見積もり精度は十分ではない。実際のレイアウトとの詳細な比較によると、誤差の最も大きな要因は

表2 実験結果

セルの種類	従来手法		提案手法		改善度	
	誤差平均	標準偏差	誤差平均	標準偏差	誤差平均	標準偏差
非 PTL セル	37.0%	33.6%	19.2%	17.9%	17.8%	15.7%
PTL セル	30.8%	24.2%	27.7%	21.6%	3.1%	2.6%
全体	35.0%	31.0%	24.8%	20.8%	10.2%	10.2%

文 献

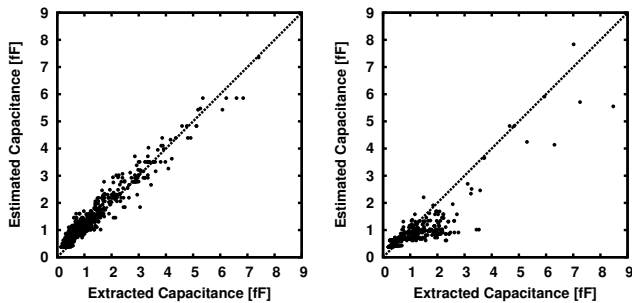


図13 抽出された寄生容量と従来手法による見積りの比較

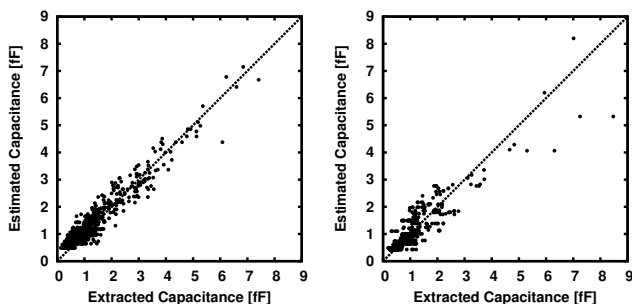


図14 抽出された寄生容量と提案手法による見積りの比較

- [1] H. Yoshida, K. De, and V. Boppana, "Accurate Pre-layout Estimation of Standard Cell Characteristics," in *Proc. of ACM/IEEE Design Automation Conference*, pp. 208–211, Jun. 2004.
- [2] J. P. Fishburn and A. E. Dunlop, "TILOS: A posynomial programming approach to transistor sizing," in *Proc. of IEEE International Conference on Computer-Aided Design*, pp. 326–328, Nov. 1985.
- [3] A. Conn *et al.*, "Gradient-Based Optimization of Custom Circuits Using a Static-Timing Formulation," in *Proc. of ACM/IEEE Design Automation Conference*, pp. 452–459, Jun. 1999.
- [4] J. L. Burns and J. A. Feldman, "C5M — A Control- Logic Layout Synthesis System for High-Performance Microprocessors," *IEEE Trans. on Computer-Aided Design*, vol. 17, no. 1, pp. 14–23, Jan. 1998.
- [5] R. Panda *et al.*, "Migration: A new technique to improve synthesized designs through incremental customization," in *Proc. of ACM/IEEE Design Automation Conference*, pp. 388–391, Nov. 1998.
- [6] M. Cote and P. Hurat, "Faster and Lower Power Cell-Based Designs with Transistor-Level Cell Sizing," in *Closing the Gap Between ASIC & Custom*, pp. 225–240, Kluwer Academic Publishers, 2002.
- [7] D. Bhattacharya and V. Boppana, "Design Optimization with Automated Flex-Cell Creation," in *Closing the Gap Between ASIC & Custom*, pp. 241–268, Kluwer Academic Publishers, 2002.
- [8] M. Guruswamy *et al.*, "CELLERITY: A Fully Automatic Layout Synthesis System for Standard Cell Libraries," in *Proc. of ACM/IEEE Design Automation Conference*, pp. 327–332, Jun. 1997.
- [9] *abraCAD Documentation*, Synopsys, Inc., 2003.
- [10] C. Bittlestone *et al.*, "Architecting ASIC Libraries and Flows in Nanometer Era," in *Proc. of ACM/IEEE Design Automation Conference*, pp. 776–781, Jun. 2003.
- [11] *International Technology Roadmap for Semiconductors 2002 Update*, Semiconductor Industry Association, 2002.
- [12] S. S. Sapatnekar *et al.*, "An Exact Solution to the Transistor Sizing Problem for CMOS Circuits Using Convex Optimization," *IEEE Trans. on Computer-Aided Design*, vol. 12, no. 11, pp. 1621–1634, Nov. 1993.
- [13] W. Liu *et al.*, *BSIM3v3.2 MOSFET Model Users' Manual*, University of California, Berkeley, 1998.
- [14] M. A. Breuer, "A Class of Min-cut Placement Algorithms," in *Proc. of ACM/IEEE Design Automation Conference*, pp. 284–290, Jun. 1977.
- [15] C. M. Fiduccia, R. M. Mattheyses, "A linear-time heuristic for improving network partitions," in *Proc. of ACM/IEEE Design Automation Conference*, pp. 175–181, Jun. 1982.
- [16] A. B. Kahng and G. Robins, "A new class of iterative Steiner tree heuristics with good performance," *IEEE Trans. on Computer-Aided Design*, vol. 11, no. 7, pp. 893–902, Jul. 1992.
- [17] M. Boehner, "LOGEX — An Automatic Logic Extractor From Transistor to Gate Level for CMOS Technology," in *Proc. of ACM/IEEE Design Automation Conference*, pp. 517–522, Jul. 1988.
- [18] R. E. Bryant, "Extraction of Gate Level Models from Transistor Circuits by Four-Valued Symbolic Analysis," in *Proc. of IEEE International Conference on Computer-Aided Design*, pp. 350–353, Nov. 1991.

配置や配線の見積もりによるものではなく、線間容量によるものと思われる。今回の手法では見積もり対象となる配線のみを考慮することで対地容量と線間容量の合計を見積もっている。対地容量については対象となる配線のみを考慮すれば十分であるが、線間容量については隣接する配線を考慮する必要がある。加えて、線間容量を対地容量とみなすことによって、シミュレーションにも誤差が生じるとと思われる。提案手法では配線をスタイナ木によって見積もっているため、これを用いて対地容量と線間容量を独立により正確に見積もることが可能であると思われる。この手法については現在検討中である。

また本論文では配線容量のみの見積もりを扱ったが、以下に示す要素もセルの性能評価もしくは最適化にとって非常に重要である。

- 配線抵抗
- 拡散の周囲長や面積
- ピン位置

これらも配置配線情報から容易に見積もることが可能であると思われる。今後はこれらの見積もり手法についても検討していく予定である。