

ULSI フロアプランニングにおける 階層的バッファブロックプランニング手法

大佐古 昌和[†] 若林 真一[†] 小出 哲士[‡]

[†]広島大学大学院工学研究科

〒739-8527 広島県東広島市鏡山一丁目4番1号

[‡]広島大学 ナノデバイス・システム研究センター

〒739-8527 広島県東広島市鏡山一丁目4番2号

E-mail:{sako, wakaba}@ecs.hiroshima-u.ac.jp, koide@sxsys.hiroshima-u.ac.jp

本稿では、チップ領域をグローバルビンに分割し、タイミングを考慮したバッファブロックプランニングを階層的に行う手法を提案する。提案手法ではバッファブロックプランニングを2段階で行う。まず第1段階では複数のネットに対するバッファブロックプランニング問題を最小コストフロー問題として定式化することにより概略的なバッファブロックプランニングを行なう。次に第2段階では、第1段階の結果に基づいて詳細なバッファブロックプランニングを行なう。また提案手法では、バッファブロックプランニングを行う前に、あらかじめバッファ配置のためのデッドスペースを生成し、3端子ネットに対するバッファ位置はネットをあらかじめ2端子ネットに変換することで決定する。

キーワード：バッファブロックプランニング、バッファ挿入、タイミング制約、独立許容領域、グローバルビン、最小コストフロー

A Hierarchical Buffer Block Planning Method for ULSI Floorplanning

Masakazu OHSAKO[†], Shin'ichi WAKABAYASHI[†] and Tetsushi KOIDE[‡]

[†]Graduate School of Engineering, Hiroshima University

4-1, Kagamiyama 1 chome, Higashi-Hiroshima 739-8527 JAPAN

[‡]Research Center for Nanodevices and Systems, Hiroshima University

4-2, Kagamiyama 1 chome, Higashi-Hiroshima 739-8527 JAPAN

E-mail:{sako, wakaba}@ecs.hiroshima-u.ac.jp, koide@sxsys.hiroshima-u.ac.jp

In this paper, we propose a hierarchical buffer block planning method, which divides the chip area into global bins, taking timing constraints into account. In this method, we execute buffer block planning in two phases. In the first phase, we formulate this problem for multiple nets as the minimum cost flow problem, and solve it globally. In the second phase, we execute detailed buffer block planning based on the result of the first phase. In the proposed method, we produce dead spaces for buffer placement before executing buffer block planning, and buffer positions of 3-pin nets are determined by transforming them into 2-pin nets.

key words: buffer block planning, buffer insertion, timing constraint, independent feasible region, global bin, minimum cost flow

1 はじめに

近年の半導体製造技術の進歩に伴い LSI の微細化が進み、配線抵抗が信号遅延に大きく影響を与えるようになってきている。また、ディープサブミクロン領域でのデザインテクノロジーにおいて、配線は回路全体のパフォーマンスや複雑さを決定する重要な要素となっている。そのため、これまでに多くの配線パフォーマンスを最適化するレイアウト設計手法が提案されており、その中でも配線経路に対するバッファ挿入は有効なアプローチの一つとして知られている [1]。

ビルディングブロック方式のレイアウト設計において高いパフォーマンスを持つ ULSI を実現するためには信号経路に多数のバッファを挿入する必要がある。配線経路へのバッファ挿入は通常、フロアプラン設計後の配線設計時に行なわれる。しかし、配線設計時に多数のバッファをブロック間に挿入するとフロアプラン設計において決定されたフロアプランの変更が必要になり、フロアプラン設計をやり直す場合が生じる。また、チップエリア内の任意の位置へのバッファ挿入を許すとチップ内の電源配線を複雑にしたり、ハード IP コアの使用が難しくなる [2]。このため、チップエリア内の一部の領域にバッファのみを集めたバッファブロックが使われるようになってきている。しかし、これまでは設計者が回路ブロック間に手作業でバッファブロックを構成していた。そこで、近年、配線設計前にあらかじめ必要なバッファを挿入し、バッファブロックを構成するバッファブロックプランニングが注目されている。

これまでいくつかのバッファブロックプランニング手法が提案されている [2, 8, 9]。Cong らは、文献 [2] においてチップエリア内に許容領域 (Feasible Region, FR) と呼ばれる領域を定義し、この領域内にバッファを挿入することでタイミング制約を満たす手法を提案した。Sarkar らは、文献 [8] において許容領域に独立の概念を加えた独立許容領域 (Independent Feasible Region, IFR) を定義し、配線可能性を考慮した手法を提案した。Tang らは、文献 [9] において許容領域を利用し、バッファブロックプランニング問題をネットワークフロー問題として定式化した手法を提案した。しかし、これらの手法はいずれも 2 端子ネットしか扱うことができないという問題点がある。

本稿ではチップエリアをグローバルピンに分割し、階層的にバッファブロックプランニングを行ない、多端子ネットも扱うことのできる手法を提案する。本手法では、2 端子ネットに対しては文献 [8] で提案された独立許容領域の概念を利用し、3 端子ネットに対しては、本手法で新たに提案する手法を用い、それ以上のネットに対してはヒューリスティック手法を用いることでバッファブロックプランニングを行なう。各ネットに対するバッファ位置は、複数のネットに対するバッファ位置割り当てを最小コストフロー問題として定式化することでグローバルに決定している。

本稿の構成は以下の通りである。まず、2. では本手法で扱うバッファブロックプランニング問題を定義し、独立許容領域の概念について述べる。次に、3. では提案手法の概要、独立許容領域の計算方法、及びネットワークフロー問題としての定式化について述べる。4. では計算機上での予備的な実験結果を示し、最後に 5. では今後の課題について述べる。

2 準備

2.1 配線遅延モデル

本稿では、配線遅延を見積もるために、配線に対しては Elmore 遅延モデル [4] を使用し、バッファはスイッチレベル RC 回路としてモデル化する。Elmore 遅延モデルは比較的精度よく配線遅延を求めることができる配線遅延モデルとして多くのレイアウト設計手法で使われている。配線遅延モデルを説明するために必要な各パラメータを表 1 に示す記号を使つて表す。

表 1 記号の定義

r	単位長あたりの配線抵抗 ($\Omega/\mu\text{m}$)
c	単位長あたりの配線キャパシタンス ($fF/\mu\text{m}$)
R_d	ドライバの駆動抵抗 (Ω)
C_l	シンの負荷キャパシタンス (fF)
T_b	バッファの固有遅延 (f_s)
R_b	バッファの駆動抵抗 (Ω)
C_b	バッファの負荷キャパシタンス (fF)

駆動抵抗 R 、負荷キャパシタンス C を持つ長さ l の配線セグメントの Elmore 遅延は以下のようになる。

$$D(R, C, l) = \frac{rc}{2}l^2 + (Rc + rC)l + RC$$

このとき、長さ L で、 n 個のバッファを持つ 2 端子ネットに対する Elmore 遅延 $D(x_1, \dots, x_n, L)$ は以下ようになる。ただし、 x_1, \dots, x_n はそれぞれ 1, \dots , n 番目のバッファ位置を表している。

$$D(x_1, \dots, x_n, L) = nT_b + D(R_d, C_b, x_1) + D(R_b, C_l, L - x_n) + \sum_{i=1}^{n-1} D(R_b, C_b, x_{i+1} - x_i)$$

2.2 問題の定式化

本稿で仮定するレイアウトモデルは矩形のブロックを配置するビルディングブロックレイアウトとし、バッファは回路ブロック間の空きスペースにのみ挿入する。また、配線遅延を計算するための配線遅延モデルとして Elmore 遅延モデルを使用する。

本稿で扱うバッファブロックプランニング問題を以下のように定式化する。

入力：フロアプラン
 ネットリスト (多端子ネットを含む)
 遅延パラメータ
 タイミング制約
 出力：各回路ブロックの配置

バッファブロック情報(大きさ, バッファ割り当て)

バッファ挿入後のネットリスト

目的: チップ面積, バッファブロック数の最小化

制約: タイミング制約

2.3 独立許容領域

ここでは, 文献[8]で提案された2端子ネットに対する独立許容領域(Independent Feasible Region, IFR)について述べる. 独立許容領域とは領域内の任意の場所にバッファを挿入することでタイミング制約を満たすことのできる最大多角形領域のことで, 同一のネット上の各バッファは独立に各領域内でバッファ位置を決定可能である. 独立許容領域の計算は以下のように行なう.

配線遅延の計算に Elmore 遅延モデルを仮定し, n 個のバッファを長さ l のネット上に最適に挿入することで配線遅延を最小にする場合の i 番目のバッファの位置は以下のようになる [1].

$$x_i^* = (i-1)y^* + x^* \quad i \in \{1, 2, \dots, n\} \quad (1)$$

ただし,

$$x^* = \frac{1}{n+1} \left(l + \frac{n(R_b - R_d)}{r} + \frac{C_s - C_b}{c} \right)$$

$$y^* = \frac{1}{n+1} \left(l - \frac{R_b - R_d}{r} + \frac{C_s - C_b}{c} \right)$$

また, このときタイミング制約を満たすために必要な最小バッファ数 k_{min} は以下のように求まる [2].

$$k_{min} = \left\lceil \frac{K_2 - \sqrt{K_2^2 - 4K_1K_3}}{2K_1} \right\rceil \quad (2)$$

ただし,

$$K_1 = R_b C_b + T_b$$

$$K_2 = T_{req} + \frac{r}{2c} (C_b - C_l)^2 + \frac{c}{2r} (R_b - R_d)^2 - (rC_b + cR_b)l - T_b - R_d C_b - R_b C_l$$

$$K_3 = \frac{1}{2} r c l^2 + (rC_l - cR_d)l + C_l R_d - T_{req}$$

ここで, T_{req} はネットのタイミング制約である.

このとき, 長さ l の2端子ネット上に(2)式で求めた k 個のバッファを挿入するとき, ネット上の i 番目のバッファに対する独立許容領域の範囲 IFR_i は以下のように表せる.

$$IFR_i = (x_i^* - W_{IFR}/2, x_i^* + W_{IFR}/2) \cap (0, l)$$

ただし,

$$W_{IFR} = 2\sqrt{\frac{D_{tgt} - D_{opt}(k, l)}{rc(2k-1)}}$$

$$D_{opt}(k, l) = D(x_1^*, \dots, x_k^*, l)$$

ここで, W_{IFR} , D_{tgt} , $D_{opt}(k, l)$ はそれぞれ独立許容領域の幅, ネットのタイミング制約, (1)式で求めた最適な場所にバッファを挿入したときの配線遅延を表している.

図1では, 1次元の独立許容領域を示している. 2次元の場合の独立許容領域は1次元の独立許容領域の和集合として定義できる [2, 8].

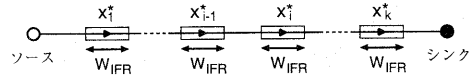


図1 1次元独立許容領域

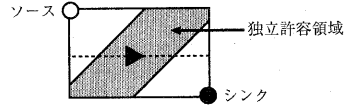


図2 2次元独立許容領域

3 提案手法

3.1 提案手法の概要

提案手法では, チップ領域をグローバルビン(図3参照)に分割し, タイミングを考慮したバッファブロックプランニングを階層的に行う. 提案手法は以下に示す3つのフェーズから成っている.

フェーズ1: 入力フロアプランに対し, タイミング制約を満たすために挿入すべきバッファ数を見積もり, その見積もりに応じてデッドスペースを調整する.

フェーズ2: チップ領域をグローバルビンに分割し, すべてのネットに対してグローバルビン単位の独立許容領域を求める. 2端子, 3端子ネットのバッファ割り当てを最小コストフロー問題として解き, 4端子以上のネットに対してはバッファ位置をヒューリスティックに決定する.

フェーズ3: 詳細な独立許容領域を計算し, フェーズ2の結果を詳細化することで, 各デッドスペース内に挿入されるバッファの詳細な配置を決定する.

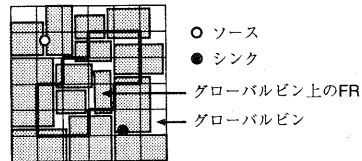


図3 グローバルビン構造

バッファ数の見積もりや, バッファブロックプランニングを行う際に, 2.3で述べた独立許容領域を計算する. 提案手法では, バッファ数を見積もるときや概略的にバッファブロックプランニングを行うときはテーブル検索によりバッファ数や独立許容領域を求め, 詳細にバッファブロックプランニングを行う際には2.3で述べた式により計算する.

以下では各フェーズの詳細について述べる.

3.1.1 フェーズ 1

従来の手法では、バッファを挿入できる領域がない場合はバッファを挿入する段階で回路ブロックをシフトし、バッファを挿入する領域を生成している。しかしこの場合、必要以上のデッドスペースが生成される可能性がある。そこでフェーズ 1 では、入力として与えられたフロアプランに対してタイミング制約を満たすために挿入すべきバッファ数を見積もり、その見積もりに応じてデッドスペースを調整することであらかじめバッファ挿入領域を確保している。

バッファ数の見積もりはネットの端子数に応じて以下の 2 つの方法で行う。まず、2, 3 端子ネットに対しては、フェーズ 2 でも使用するテーブルを使うことで、高速にバッファ数を見積もる。4 端子以上のネットに対しては、文献 [7] や文献 [3] のような高速なヒューリスティック概略配線手法を用いることでバッファ数を見積もる。

次にデッドスペースの調整は以下のようにして行う。まず、提案手法では、デッドスペースを作るために個々のブロックを動かしてデッドスペースを均等に作るのではなく、ある大きさにクラスタリングしたブロック集合を動かすことでデッドスペースを作る。これは、バッファブロックへの電源配線等を考慮して、バッファブロックとして使用するデッドスペースにはある程度の大きさを確保すると共に、チップ全体でのバッファブロックの総数があり大きくなりすぎないようにするためである。

3.1.2 フェーズ 2

提案手法では、階層的にバッファブロックプランニングを行うために、チップ領域を正方形のグローバルビンに分割する。フェーズ 2 では、各ネット上の各バッファを挿入するグローバルビンに決定する。そのとき、グローバルビンへのバッファの割り当てを最小コストフロー問題として定式化することで、概略的なバッファブロックプランニングを行う。

フェーズ 2 では、以下のようにしてグローバルビンへのバッファ割り当てを決定している。まずチップ領域全体をグローバルビンに分割し、各ネットに対してグローバルビン単位の独立許容領域を計算する。そこで、グローバルビンを 1 行または 1 列スキャンし、スキャンした領域に独立許容領域が含まれるネットに対してバッファ割り当てを決定する。このとき、バッファ割り当てを最小コストフロー問題として定式化することで複数のネットに対するバッファ割り当てを同時に決定する。この割り当て問題を最小コストフローパッケージ MCF[5] を使ってこの問題を解くことでバッファの概略的な割り当てをグローバルに決定する。ある行または列での割り当て問題を解くと、スキャン領域は例えば左から右、上から下のように一定方向に移動する。ネットワークフローによる定式化の詳細は 3.3 で述べる。

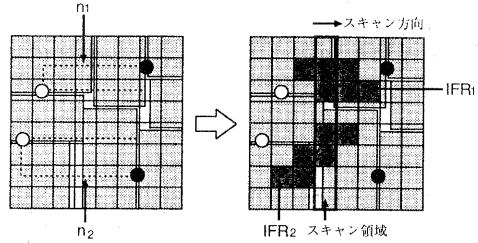


図 4 独立許容領域のスキャン

3.1.3 フェーズ 3

フェーズ 3 では、フェーズ 2 で決定されたバッファ割り当てを詳細化する。まず、グローバルビン内のデッドスペースを矩形チャネル構造 (図 5 参照) に分割し、詳細な独立許容領域を計算する。そこで、フェーズ 2 の結果と詳細な独立許容領域を考慮し、バッファ割り当てを調整する。この調整とは、バッファを近隣のグローバルビンに割り当て直すなどの操作を意味している。最終的な割り当てが決定すると、各グローバルビンに割り当てられたバッファを 2 段階で詳細化する。まず、1 段階目では、デッドスペースから得ることのできるチャネル構造上にバッファを割り当て、割り当てられたバッファ集合をバッファブロックとする。2 段階目では、バッファブロックに割り当てられたバッファの最終配置を決定する。

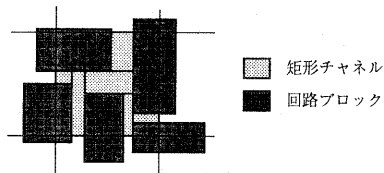


図 5 矩形チャネル構造

3.2 独立許容領域の計算

ここでは、独立許容領域の計算方法について述べる。提案手法では、独立許容領域は (1) 2 端子ネット、(2) 3 端子ネット、(3) 4 端子以上のネット、の 3 つの場合に分けて計算する。

まず、2 端子ネットに対する独立許容領域は 2.3 で述べた方法で計算する。このとき、同じ計算を繰り返すことによる計算時間の増大を抑えるためにバッファ数を見積もるとき、概略的なバッファブロックプランニングを行なうときは、テーブル検索によってこの計算を行なう。このテーブルは、入力として配線長、タイミング制約、駆動抵抗、負荷キャパシタンスを与え、出力として挿入バッファ数、各バッファに対する独立許容領域の中心座標と大きさを出力する。

3 端子ネットに対する独立許容領域の計算は以下

のようにして行う。3端子ネットの独立許容領域も2端子ネットと同様の方法で処理するために、スタイナポイントを含み、バッファに囲まれている3端子ネットを図6のように、総配線長が等しくなる2端子ネットとして扱う。このとき、最終的に求めるのはソース-シンク1間の配線遅延とする。これより、3端子ネットの配線遅延は配線長が同じ2端子ネットの配線遅延を用いて表すことができるので、3端子ネットを2端子ネットと同様に扱うことができる。

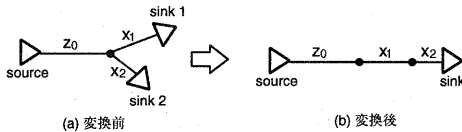


図6 3端子ネットの変換

これを実際の3端子ネットに適用する。まず、バッファ位置の決まっていない3端子ネットが与えられたとする。そこで、スタイナポイントからシンク2方向に長さ x_2 の配線セグメントを仮定する。この配線セグメントをシンク1側に移動することで図6のように2端子ネットに変換できる。よって、(1)式により図7(b)のような X, Y, Z を求めることができ、独立許容領域を求めることができる。このとき、 z_0 をソースからスタイナポイントへの配線セグメントでの最後のバッファからスタイナポイントまでの長さ、 x_1 をスタイナポイントからシンク1への配線セグメントでのスタイナポイントから最初のバッファまでの長さとし、 $Y = z_0 + x_1 + x_2$ が成り立っていると仮定する。また、配線セグメント x_2 を含めた2端子ネットの長さを L 、挿入バッファ数を k とすると、 $Z = L - X - (k-1)Y$ も成り立っている。この結果をソースからスタイナポイント方向、シンク1からスタイナポイント方向に適用し、残りをスタイナポイントからシンク2方向に適用する。

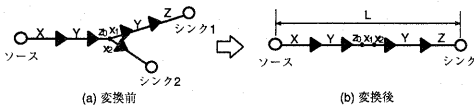


図7 3端子ネット

4端子以上のネットに対しては、独立許容領域を用いるのではなく、バッファを挿入することでタイミング制約を満たすことのできるグローバルピンを見つけることで、独立許容領域の計算と等価な処理を行なったとみなす。このとき、文献[3]のようなバッファ位置を考慮した概略配線アルゴリズムを簡単化したものを使用することで、バッファ数やバッファ位置を見積もる。

3.3 最小コストフローによる定式化

提案手法では、グローバルピンへの概略的なバッファ割り当てを同時に決めるために、バッファ割り当て問題を最小コストフロー問題として定式化している。ここで、最小コストフロー問題とは、問題に依存したフローグラフが与えられ、グラフ内に一定量のフローを流すとき、コスト和が最小となるようなフローを求める問題である。このとき、フロー上の各枝はコストと容量を持っている。

提案手法におけるフローグラフは図8(b)に示されているようなものを使用する。このフローは、ソースノード s 、ターゲットノード t 、ネットノード n_i 、バッファノード b_j により構成されている。ソースノードからネットノードへの枝は、スキャン領域にバッファを割り当てることのできるネットを表しており、枝には容量1とコスト c_i を与えている。コスト c_i は、ネット i に対してスキャン領域に含まれている独立許容領域を表すグローバルピン数とまだスキャンされていない独立許容領域を表すグローバルピン数の和とする。ネットノードからバッファノードへの枝は、ノード i がグローバルピン j にバッファを割り当てることができるということを表しており、枝には容量1とコスト0を与えている。バッファノード同士をつなぐ枝は、グローバルピン j にバッファを割り当てることができることを表しており、枝には容量 b_j とコスト0を与えている。容量 b_j は、グローバルピン j に挿入することのできるバッファ数を表している。

例を使って説明する。例えば、図8(a)のように、あるスキャン領域に独立許容領域が含まれているネット n_1, n_2, n_3 上のバッファをグローバルピン b_1, b_2, b_3, b_4 に割り当てることができることを考える。このとき、スキャン領域内の各ネットに対する独立許容領域を表すグローバルピン数はそれぞれ2, 3, 1個である。また、まだスキャンされていない独立許容領域のグローバルピン数をそれぞれ1, 3, 0個とすると、各ネットのコストはそれぞれの和、すなわち $c_1=3, c_2=6, c_3=1$ となる。このとき、ネット n_3 のコストが小さいので、割り当てられやすくなる。これに対して、フローグラフは図8(b)のようになる。

4 実験の評価

提案手法は現在プログラムとして実現中であるが、ここではフェーズ2のみC言語で実現し、UltraSPARC-II(300MHZ)プロセッサ上でシミュレーション実験を行った結果を示す。今回の実験では、グローバルピンのスキャンはある行に対して、左から右方向にのみ行ない、入力フロアプランに含まれているデッドスペースのみをバッファブロックとして使用した。実験で使用した遅延パラメータを表2に示す。実験で用いたデータの詳細を表3に示す。ただし、表3中の $\#block$ はブロック数、 $\#net$ はネット数、 $\#2pinnet$ は2端子ネット数、 A_{lb} は面積下界値 $[mm^2]$ を表している。また、入力フロアプランは当研究室で開発したタイミングを考慮したフロアプランニング手法[6]の結果から得た。

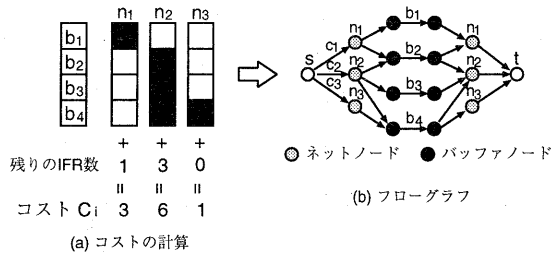


図 8 フローグラフの例

表 2 遅延パラメータ

単位配線抵抗 r ($\Omega/\mu\text{m}$)	0.075
単位配線キャパシタンス c ($fF/\mu\text{m}$)	0.118
ソースの駆動抵抗 R_d (Ω)	180
シンクの負荷キャパシタンス C_i (fF)	23.4
バッファの固有遅延 T_b (ps)	36.4
バッファの駆動抵抗 R_b (Ω)	180
バッファの負荷キャパシタンス C_b (fF)	23.4

表 3 テストデータ

データ	#block	#net	#2pinnet	A_{1b}
a.m46	46	2564	2007	24.93
b.m84	84	5830	5127	30.07

実験結果を表 4 に示す。ただし、表中の #need は タイミング制約を満たすために必要なバッファ数の総和、#buffer は実際に挿入されたバッファ数、#block はバッファブロック数、#MCF は最小コストフローパッケージ MCF の呼び出し回数、#CPU は計算時間 [sec] を表している。この結果より、データ a.m46 に関しては、必要なバッファはすべて挿入され、データ b.m84 に関しては必要なバッファの 89% が挿入されており、フェーズ 2 の有効性を確認することができる。しかし、データ b.m84 は入力フロアプランのデッドスペースだけではバッファ挿入領域が不足しているため、デッドスペースを適切に調整する操作、すなわちフェーズ 1 の操作が必要であることがわかる。また、バッファ数に対して、バッファブロック数が多いので、バッファブロックの最適配置が必要であることもわかる。このことから、バッファ割り当ての段階で、個々のバッファブロックに含まれるバッファ数を大きくできるようにフローグラフのコスト関数を工夫することが必要であると思われる。

表 4 実験結果

データ	#need	#buffer	#block	#MCF	#CPU
a.m46	103	103	71	27	11.4
b.m84	398	355	234	51	60.5

5 あとがき

本稿では、階層的にバッファブロックプランニングを行なう手法を提案した。提案手法ではまず概略的なバッファブロックプランニングを行ない、その結果に基づいて詳細なバッファブロックプランニングを求める。前者に対しては最小コストフローに基づくグローバルな解探索を実現し、質の高い解を短時間で求めることが可能である。今後の課題としては、手法全体の実現、および大規模実データを用いた提案手法の実験的評価等が挙げられる。

謝辞：本研究の一部は、文部科学省科学研究費補助金基盤研究 (B)(2) (課題番号 12555097) による。

参考文献

- [1] C. Alpert and A. Devgan: "Wire segmenting for improved buffer insertion", Proc. of 34th ACM/IEEE Design Automation Conference, pp.588-593, 1997.
- [2] J. Cong, T. Kong and D.Z. Pan: "Buffer block planning for interconnect-driven floorplanning", Proc. IEEE/ACM International Conference on Computer-Aided Design, pp.358-363, 1999.
- [3] J. Cong and X. Yuan: "Routing tree construction under fixed buffer locations", Proc. of 37th ACM/IEEE Design Automation Conference, pp.379-384, 2000.
- [4] W. C. Elmore: "The transient response of damped linear network with particular regard to wideband amplifier", J. Applied Physics, pp.55-63, 1948.
- [5] "MCF - A network simplex implementation", Konrad-Zuse-Zentrum fuer Informationstechnik Berlin, <http://www.zib.de/Optimization/Software/Mcf/index.html>
- [6] S. Nakaya: "A performance-driven ULSI floorplanning method with a table lookup technique for efficient interconnect estimation", Master thesis, Graduate School of Engineering, Hiroshima University, 2001.
- [7] T. Okamoto and J. Cong: "Buffered Steiner tree construction with wire sizing for interconnect layout optimization", Proc. of International Conference on Computer Aided Design, pp.44-49, 1996.
- [8] P. Sarkar and C.-K. Koh: "Routability-Driven Repeater Block Planning for Interconnect-Centric Floorplanning", IEEE Trans. on Computer-Aided Design of Integrated Circuits and System, Vol.20, No.5, pp.660-671, 2001.
- [9] X. Tang and D.F. Wong: "Planning buffer locations by network flows", Proc. of International Symposium on Physical Design, pp.180-185, 2000.