

配線長を考慮したクラスタリングに基づく スタンダードセル階層配置手法

窪田 創[†] 若林 真一^{††} 小出 哲士^{†††} 辻 敏夫[†]

[†] 広島大学大学院工学研究科 〒739-8527 広島県東広島市鏡山1丁目4番1号

^{††} 広島市立大学情報科学部 〒731-3194 広島県広島市安佐南区大塚東3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1丁目4番2号

E-mail: †{ippo,tsuji}@bsys.hiroshima-u.ac.jp, ††wakaba@computer.org,

†††koide@sxsys.hiroshima-u.ac.jp

あらまし 近年、半導体技術の急激な進歩に伴い、回路全体のパフォーマンスを決定する要素として配線遅延が重要となってきている。そこで、配線長の短縮を考慮したセルのクラスタリングを行って問題サイズを減少させ、クラスタ単位でセル配置を決定する階層配置手法が一般的に用いられている。従来の手法ではクラスタ単位の配置の際にクラスタ内のネット情報を利用していなかったが、本稿では階層設計においてクラスタ内のセル配置を考慮しながらクラスタ配置を行う手法を提案する。提案手法では最終配置において近隣に配置されることが望ましいセルをクラスタ配置時に考慮するために、クラスタ内のセルに対して中心セル、境界セル、非境界セルなどのセル集合に分類する。このようにすることで、提案手法は階層設計において後の階層で決定されるセル配置を前もって予測することが可能になり、配線長の短い優れたセル配置を生成することができる。

キーワード 階層配置, クラスタ, 中心セル, 境界セル, 非境界セル

A Hierarchical Standard-Cell Placement Method Based on Wire Length-Driven Clustering

Hajime KUBOTA[†], Shin'ichi WAKABAYASHI^{††}, Tetsushi KOIDE^{†††}, and Toshio TSUJI[†]

[†] Graduate School of Engineering, Hiroshima University

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

^{††} Faculty of Information Sciences, Hiroshima City University

3-4-1, Ozuka-higashi, Asaminami-ku, Hiroshima, 731-3194 Japan

^{†††} Research Center for Nanodevices and Systems, Hiroshima University

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

E-mail: †{ippo,tsuji}@bsys.hiroshima-u.ac.jp, ††wakaba@computer.org,

†††koide@sxsys.hiroshima-u.ac.jp

Abstract Due to the rapid progress of semiconductor technology, wiring delay becomes important on determining the performance of whole circuit. So, wire length-driven hierarchical placement methods are generally used, in which clusters consisting of sets of cells are generated. While previous methods determine clusters without considering internal net information in the cluster, in this paper, we present a cluster placement method considering internal cell and net information in the cluster in the hierarchical placement. When we determine cluster placement, we consider the cells to be placed closely. So, we identify cells in the cluster as center-cells, boundary-cells and unboundary-cells. It makes it possible to predict cell placement in the following placement stage, and the proposed method can produce good cell placement with shorter wire length.

Key words hierarchical placement, cluster, center-cell, boundary-cell, unboundary-cell

1. ま え が き

半導体技術の進歩に伴い、1チップに搭載されるゲート数が数千万である大規模集積回路が登場してきた。本稿ではこのような大規模VLSIチップのスタンダードセルレイアウト設計における配置手法に注目する。従来の配置手法は大まかに次の3種類のアプローチに分類できる。すなわち、ミニカット手法と呼ばれるグラフ分割に基づく手法[2]や、線形計画法などの数理計画法に基づく手法[3]、さらには反復的な改良を行うことで徐々に最適化を計る手法[4]などがある。これ以外にも、これらの手法を組み合わせた手法[1]など、数多くの手法が知られている。

しかしながら、回路規模の増大により、与えられたネットリストをそのまま扱うことはCPU時間とメモリ量の制約から困難となってきたため、セルをクラスタリングすることにより問題サイズを減少させ、クラスタ単位で配置する階層配置手法が一般的に用いられるようになってきた。一方、回路のパフォーマンスを決定する要素として配線長、遅延時間、チップ面積、配線混雑度など様々な要素がある。近年のディープサブミクロン超微細加工技術においては回路遅延で支配的となるのはゲート遅延ではなく、配線遅延となる。配線遅延は配線長の2乗に比例するので、配線長を最小化することが遅延時間の短縮につながる。また、配線混雑度も回路のパフォーマンスを決定する重要な要素であるが、配線長を変化させることなく混雑度を和らげる手法[5]などが提案されており、こういった手法を用いると配線長と混雑度を同時に考慮する必要がなくなる。このため、本稿では配線長の最小化に注目した配置手法を提案する。

クラスタリングに基づく階層的配置手法において、配線長を最小化する最終的な配置結果を効率よく得るためにはクラスタリングにおいて配線長を考慮する必要がある。一般にこのクラスタリングは配置の前工程で行うため、セル配置の情報を利用することはできない。そこで、2端子ネットのような端子数の少ないネットの端子となるセルが近隣に配置される傾向にあることに着目して、端子数の少ないネットの端子を優先的に同一のクラスタに取り込む手法[6]などが提案されている。しかし、必ずしも全てのネットの端子に対して近隣に配置される保証はないという欠点がある。また、このクラスタを配置していく際に[1]などの手法は問題の複雑さを回避するためにクラスタ内のセル間のネット情報を利用せずにクラスタ間のネットの情報のみで配置を行っている。しかし、実際はクラスタ配置の際にクラスタ内のセルの交換などによるクラスタの改良を随時行っているため、最適なクラスタ配置はクラスタ自体の変化のために常に変動する可能性がある。

そこで本手法では近傍に配置されるべきセルはセルにつながっているネットの端子数ではなくネットの深さに基づいて選択すべきであると考え、クラスタの中心位置に配置されるセルを予測して中心セルとして、この中心セルからのネットの深さを浅くするクラスタリングを採用している。さらに、各階層でクラスタを配置する際にクラスタ内の情報を利用することで後の工程でクラスタ内のセルが配線長を最小化するために移動さ

せる方向も前もって考慮できるような工夫も採り入れている。

2. 準 備

2.1 ネット距離

数百万ものセルに対して同時に最適な配置を行うことは配置パターンが膨大になることから不可能である。このため、配置の前段階から配置対象であるセルの数を減少させるクラスタリングと呼ばれる方法が用いられている。理想的には最適配置において近隣に配置されるセル同士を同一のクラスタにすることが望まれるが、実際の配置手法においてはセル配置の前にクラスタを生成する必要がある。このため、セル配置前の段階において実際に配置した後に近隣に配置されるセルを予測する方法が研究されてきた[6]。これは2端子ネットなどの端子数の少ないネットの配線長を優先的に短くすることが総配線長を短くするには有効的であるということから、端子数の少ないネットにつながるセルを同一のクラスタに取り込む方法である。しかし、この方法では1つのクラスタに含まれるセルが鎖状に選択される可能性があり、鎖の両端になるセル同士が最適なセル配置においては近くに配置される保障はないという問題点がある。

そこで、本提案手法ではネットの本数ではなくどれだけのネットを経由しているかに注目する。

[仮説1] セル α からセル β までのネット距離 $NL(\alpha, \beta)$ をセル α から幅優先探索によりネットを経由してセルを探索した際に初めてセル β に到達するパスにおいて通過したネットの本数と定義する。ネット距離が短いセル同士ほど、最適なセル配置において、近接して配置されるものと仮定する。

例：セル α とセル β があるネットで直接つながっている場合は $NL(\alpha, \beta) = 1$ となる。

2.2 概略配置における評価関数

従来の階層配置手法では概略配置においてクラスタ間のネットの配線長を最小化するにあたり、問題を解く複雑さを減少させるためにクラスタ内の情報をブラックボックスとして扱ってきた[1], [2]。しかし、次の階層においてクラスタを分解して配置を行う際に現在の最適なクラスタ配置がそのまま保持される保証は一般的にはない。これは、クラスタ配置においてクラスタ内の情報を無視していることが原因となっている。

そこで、本手法ではクラスタを配置する際にクラスタ内のセルの接続関係を考慮して、クラスタ間のネットの配線長を用いて2端子ネットの配線長に関して次に示す概略配置における新しい評価基準(仮説2とする)を設ける。

[仮説2] クラスタAの端子 a とクラスタBの端子 b が2端子ネットにつながっているとす。この時、クラスタAの中心サブクラスタ a_0 から端子 a までのネット距離 $NL(a_0, a)$ と、クラスタBの中心サブクラスタ b_0 から端子 b までのネット距離 $NL(b_0, b)$ を用いて、このネットの長さ $VN(a, b)$ を式1のように表す。なお、式1中の L_a は $NL(a_0, a)$ 、 L_b は $NL(b_0, b)$ を略したものであり、 HN は端子 b と端子 a を囲む矩形の半周近似の値を示している。最適なクラスタ配置はこのネット長を最短にすることに得られるものと仮定する。

$$VN(a, b) = \begin{cases} HN/(L_a + L_b) & (L_a < 2 \& L_b < 2) \\ HN/(2 + L_b) & (L_a \geq 2 \& L_b < 2) \\ HN/(L_a + 2) & (L_a < 2 \& L_b \geq 2) \\ HN/(2 + 2) & (\text{上記以外}) \end{cases} \quad (1)$$

なお、式中の定数 2 は実験的に定めている。

また、多端子ネットの配線長に関してはネットにつながる全端子の重心位置に仮想的な端子を設け、各ネットの端子がこの仮想的な端子に 2 端子ネットとしてつながるというスター配線を用いることで式 1 を使用している。最終的に得られたセル配置の配線長の評価の際には多端子ネットに関しては全端子を囲む最小矩形の半周近似を用いているので、スター配線で見積もった配線長と半周近似で見積もった配線長が近づくように評価式を工夫している。

3. 提案手法

本提案手法は大規模回路を扱うためにセルのクラスタリングに基づいて階層的に配置を行う。提案手法全体のフローを図 1 に示す。

提案手法の各工程の概略を以下に示す。まず、入力データであるセル集合に対して 2 段階のクラスタリングによりサブクラスタ、クラスタ集合を生成する。このクラスタリングの工程では従来のクラスタ間のネット数を少なくするようなミニカットパーティショニング [2] などとは異なる方法を採用している。詳細は 3.1 節にて説明する。次にクラスタを配置する概略配置を行う。ここでは SA (シミュレーティッドアニーリング) を使用してクラスタを階層的に配置していく。各階層を経る毎にクラスタのサイズを減少させ、必要に応じてサブクラスタを分解することで詳細なセル (サブクラスタ) の配置位置を決定していく。この概略配置工程においても配置時の評価としてクラスタ間のネットの配線長ではなく、仮説 2 に基づいてクラスタ内のネットの情報を利用した配置を実現している。クラスタのサイズの縮小とクラスタの配置を交互に繰り返してクラスタのサイズが十分小さくなった時点で詳細配置に移る。この詳細配置ではクラスタに属しているセルをクラスタを配置している領域内のセル行と呼ばれるセルの配置領域にセルを配置していく。以降では各工程の詳細を説明する。

3.1 クラスタリング

クラスタリングの目的は問題サイズの減少と最終的な配置を想定して際に互いに近くに配置されると考えられるセルのペアを発見し、そのようなセルは同じクラスタにクラスタリングすることである。ここで生成する 2 種類のセル集合であるサブクラスタとクラスタについて以下に説明する。

【サブクラスタ】 まず、セル同士でのクラスタリングを実現する。ここでは式 2 に示す評価基準を用いてサブクラスタを生成している。

$$net_{in}/(net_{in} + net_{out}) < \alpha \quad (2)$$

ここで、 net_{in} はサブクラスタに含まれるセル集合につながっているサブクラスタ外部に出ないネット数で、 net_{out} はサブクラ

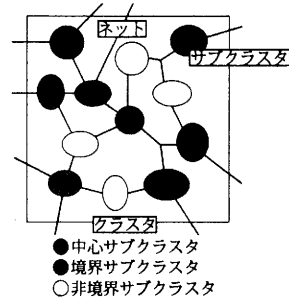


図 2 クラスタを構成しているサブクラスタ

スタ外部に出るネット数である。さらに、 α は定数であり、サブクラスタ数をセル数の $2/3 \sim 3/4$ となるように調節している。

【クラスタ】 1 個のクラスタは上で作成したサブクラスタとセルの集合で構成される。提案手法では総配線長を最小化した最適配置において近隣に配置されるセルを同一のクラスタに取り込むことを目的として、仮説 1 のネット距離の概念を用いて配置する前の段階から実際に配置を行う場合に近隣に配置されるセルを予測することでクラスタリングを実現している。クラスタは以下の 3 種類のサブクラスタから構成される (図 2 参照)。

【中心サブクラスタ (中心セル)】 クラスタ内の中心位置に配置されると想定されるサブクラスタ (セル) である。

【境界サブクラスタ (境界セル)】 他のクラスタとネットを介してつながっているサブクラスタ (セル) であり、クラスタの外周部に配置されると考える。

【非境界サブクラスタ (非境界セル)】 クラスタ内のサブクラスタの内、中心サブクラスタや境界サブクラスタ以外のサブクラスタ (セル) のことを示す。このサブクラスタから外部のクラスタへ直接つながるネットを持たないため、中心サブクラスタの近くに配置されると考える。

各クラスタにおいて非境界サブクラスタはクラスタ外部へのネットを持たないため、クラスタの中心近くに配置することが望まれる。逆に境界セルはクラスタを配置する際に他のクラスタとの配置関係を決定する重要な要素であり、従来研究ではこの境界セルが少なくなるようなクラスタを実現している。本提案手法では、この境界セルを単純に減少させるのではなく仮説 1 のネット距離の概念を用いてクラスタに含まれるサブクラスタが中心サブクラスタからのネット距離が短く、なおかつ境界セルが少なくなるようなクラスタリングを行っている。すなわち、本手法では中心サブクラスタからのネット距離が大きいサブクラスタ (セル) はクラスタの中心から配線長が長いという仮定を設けている。この仮定により、従来手法のクラスタリングにおいては境界セル数が少なくなるように、かつ互いに配線長が短くなるサブクラスタ集合をクラスタに取りこんでいくことで、結果として最適なクラスタ配置においては近接して配置されることのないサブクラスタ同士を同一クラスタの要素として取りこんでしまうことを避けている。

3.2 概略配置

この概略配置の工程ではクラスタの配置、クラスタの分割

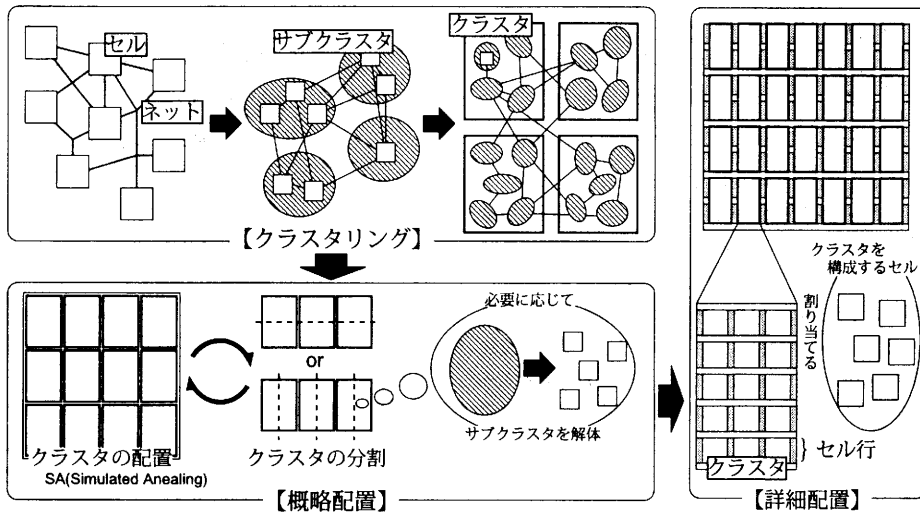


図1 提案手法のフロー

を交互に反復して行っている。クラスタは最初はサブクラスタの集合として扱っているが、クラスタを分割するに従い必要に応じてサブクラスタをセルの集合に分解している。さらに、SA(Simulated Annealing)を用いてクラスタ内部のサブクラスタ(セル)のネットの接続関係も考慮して配置を行う。通常の逐次探索手法は局所解へ収束する欠点を持つのにに対し、SAは解の質と探索回数に依存した受取確率を用いることで局所解への誤った収束を避けることを可能にした手法である。SAは逐次探索手法に比べて探索空間が広がるため実行時間が長くなる欠点を有するがパフォーマンスの良い結果を得ることが可能である。また、各クラスタのサイズが小さくなり、クラスタの数が多くなるとこの工程の探索数が非常に大きくなるため、SAに基づく概略配置ではセルの最終配置までを定めることはせず、次の詳細配置において最終的なセルの配置位置を決定する。配置前のクラスタリングでは利用可能なセル配置情報が存在しないため、配置情報を利用することなくクラスタを作成したが、概略配置の途中の段階においては、クラスタの配置は求まっているので、このクラスタの配置情報を利用して近隣に配置されているクラスタ同士で再クラスタリングを行うことで配置情報を利用したクラスタを作成することが出来る。以下ではこの概略配置の流れを詳細に説明する。

【クラスタの配置】 この工程の目的はクラスタの配置位置を決定することである。チップ領域をクラスタの数に合うように格子状に分割し、各格子にクラスタを割り当てることで配置を実現する。最初に配置するクラスタの数を2としているため、クラスタの数と格子数が一致することができるようにしている。なお、配置の改良はクラスタのランダムな交換により実現しており、ここでは解の受取判定にSAを用いる。SAには解の探索数を制御したり改善解の受取確率を決定するパラメータとして冷却係数 $Cool$ 、内部ループ数 $Loop$ 、温度 $temp$ 、初期温度 $temp_{ini}$ 、そして終了温度 $temp_{fin}$ というパラメータを持つ。探

索数、改善解に対する受取確率は次式で計算される。

$$\text{探索数} = Loop \times \lceil \log_{Cool} \left(\frac{temp_{ini}}{temp_{fin}} \right) \rceil$$

$$\text{受取確率} (\%) = 100 \times \exp \left(-\frac{Cost' - Cost}{temp} \right)$$

ここで使用している $Cost$ は解の改良前の評価値、 $Cost'$ は解の改良後の評価値である。受取確率計算中で使用している温度 $temp$ は $temp_{ini}$ から開始し、 $Loop$ 回探索を行う毎に $Cool$ が掛けられて最終的に $temp_{fin}$ となった時点で解の探索を終了する。本手法では各階層で配置した情報がある程度次の階層で保持できるように階層毎に各種パラメータを変動させるようにしている。ここでは次の【クラスタの分割】を行った際に次の階層の配置を行うとする。そこで、今回は以下のように各パラメータを設定している。

《冷却係数》 各階層に対しても0.9に統一。

《内部ループ数》 (クラスタ数 < 100) の場合はその階層におけるクラスタ数の2乗、(クラスタ数 ≥ 100) の場合はクラスタ数 $\times 100$ を使用している。

《初期温度》 直接、温度を設定するのではなく、各階層において始めに内部ループ数回配置の改良を試して得られた受取確率を参照しながら温度を2倍あるいは $\frac{1}{2}$ 倍することで調整している。なお、試みに改良した配置は実際には採用しておらず、実際の配置の改良は試みに配置改良を行う前の状態から行うものとする。この操作により受取確率が0.7になるように初期温度を設定してからSAを開始し、階層を経るごとにSAの初期温度を0.05ずつ減少して最終的には0.2まで下がったところで初期温度を固定するようにしている。

《終了温度》 こども初期温度と同様に受取確率を参照しながら終了温度を設定している。本稿では各階層における終了温度は受取確率が0.05となる温度で統一している。

これらのパラメータは実験的に実行時間と最終的な解のトレードオフを検討して定めた値を用いている。

【クラスタの分割】ここでは配置したクラスタを縦あるいは横方向に2分割する工程である。前工程である【クラスタの配置】の際に仮説2で説明した評価値を使用しているが、この評価値は次の条件の際にそのネットの配線長が最小となる値を予測したものである。その条件はクラスタ間のネットのパス上に各クラスタの中心サブクラスタからクラスタの端子となる境界サブクラスタまでに通過するサブクラスタが均等に分布する場合である。本工程ではこの条件を満たすようにサブクラスタを移動させることを目的としている。そこで、クラスタ間のネットに着目してネットの端子を囲む最小矩形の半周長を配線長とする半周近似による配線長と本手法の配置の工程で使用した評価値を比較して、評価値が大きいネットの内でおおかつ半周近似による配線長が長いネットの端子となる境界サブクラスタを優先的に移動するようにクラスタを2分割していく。

【再クラスタリング】一度、概略配置工程でクラスタを細分化して詳細なクラスタの配置を得たら、次段階である詳細配置に移る前に、配置情報を利用したクラスタリングを行うために、もう一度、クラスタリングを行う。3.1節で行ったクラスタリングでは配置情報を一切使用していないため、この工程では配置情報を利用したクラスタリングを行うことで最終的なレイアウトにおいて近隣に配置されるセル集合を同一のクラスタにするということを実現する。配置のパターンが頻繁に変動する最上位の配置の階層に戻るのではなくある程度、現在の配置情報を保存できるように100~200個のクラスタを作成することにして、再度概略配置を行う。

3.3 詳細配置

ここでは概略配置で得られたクラスタの配置に対してクラスタに属するセルをクラスタが配置されている領域に割り当てていく。紙面の都合上、詳細は省略するが、クラスタ間のネットの配線長最小化を目的とした逐次的な配置手法で実現している。

4. 実験結果

提案手法は現在開発途中であり、本手法の有効性を検証し、手法の改善点を考察するために、計算機実験を行った。

計算機実験においては2種類のベンチマークデータを用いた。すなわち、各セルのサイズが均一で、最適解が公開されているベンチマークデータ（以下ではPekoデータとよぶ）[7]と、セルのサイズにばらつきがあり、最適解は未知であるベンチマークデータ（以下ではIBMデータとよぶ）[8]を使用している。また、比較手法として以下の2種類の配置手法を使用している。

【Dragon】トップダウン的にクラスタリングを行い、ボトムアップ的にクラスタの配置位置を決定していく階層配置手法。URL [9] からダウンロードしたものを使用している。

【Capo】パーティショニングを利用した階層配置手法。URL [10] からダウンロードしたものを使用している。そして実験環境としていずれの手法もIntel(R) Pentium(R)4 CPU 2.26GHzのワークステーション上で実験を行っている。

表1はPekoデータに対する実験結果である。各データに対して比較手法であるDragon、Capoより短い総配線長が得ら

表1 Peko(site1) データに対する実験結果

回路名 (セル数)	提案		Dragon		Capo	
	PW	CPU(s)	PW	CPU(s)	PW	CPU(s)
01(12506)	1.54	608	1.79	743	2.24	25
02(19342)	1.64	1074	1.93	1276	2.22	43
03(22853)	1.45	1384	1.95	1425	2.28	54
04(27220)	1.33	2730	2.21	2687	2.28	65
05(28146)	1.62	3663	1.98	4027	2.27	71
06(32332)	1.42	3896	2.11	3219	2.35	81
07(45639)	1.29	5526	2.17	2287	2.44	120
08(51023)	1.45	6990	2.16	5847	2.27	138
09(53110)	1.36	8431	2.12	4851	2.38	154
10(68685)	1.46	10951	1.79	6445	2.42	210
11(70152)	1.67	13144	1.94	5370	2.38	207
12(70439)	1.58	14310	1.93	7287	2.50	226
13(83709)	1.67	15640	2.15	7072	2.36	270
14(147088)	1.70	17971	2.01	9347	2.42	483
15(161187)	1.59	18823	2.17	12693	2.40	623
16(182980)	1.47	20230	2.26	14077	2.50	698
17(184752)	1.68	22401	2.50	28308	2.53	744
18(210341)	1.79	25695	2.38	25891	2.53	779

(表中のPWは最適な総配線長に対する比率(%))

れている一方、実行時間に関してはDragonとあまり違いが見られずCapoに対しては40倍ほど時間が掛かっている結果が得られた。実行時間が多大であることはSAを用いて多大な探索空間を探索していることが要因である。また、配線長に関して比較手法に比べて優良な結果が得られた点を考察する。このPekoデータはネットの端子が必ず近隣に配置されるということを保証している回路である。従来のクラスタリングではクラスタ間のネットの数を最小化する方法が採用されているが、クラスタを配置する際に近隣配置するクラスタに対してはクラスタ間のネットの本数が多くても問題はなく、従来の方法ではこういったことを実現できなかった。しかし、本手法においては各クラスタに対して中心サブクラスタ(セル)というものを設け、このサブクラスタからネット距離が短くなるようなクラスタリングを行っているため、クラスタ間のネットの本数を最小化することが目的ではなく各クラスタの中心サブクラスタからネット距離が短くなることを優先していることが効果的に働いたと考えられる。また、概略配置の際にもクラスタのネットの端子となるサブクラスタ(セル)がより近くに配置されるような評価基準を用いていることの効果も少なからず作用していると考えられる。

次に表2に示すIBMデータに対する実験結果について考察する。この表では表1とは異なり、比較手法に比べて配線長が大幅に悪くなっている。実行時間に関しては表1と大きな差はないが、配線長については最悪で2.63倍も比較手法に比べて悪くなっている。以下ではIBMデータに対して提案手法の配線長が大幅に悪い要因を考える。Pekoデータが最適解を既知とするために人工的に作成されたデータであるのに対し、このIBMデータは実データに基づくベンチマークデータである。このため、IBMデータでは部分回路や異なるサイズのセルが混在し

表 2 IBM データに対する実験結果

回路名 (セル数)	提案		Dragon		Capo	
	PW	CPU(s)	PW	CPU(s)	PW	CPU(s)
01(12520)	2.63	608	1.00	781	1.20	30
02(19580)	1.62	1214	1.00	1203	1.12	66
07(45385)	1.48	2154	1.00	1926	1.14	182
08(51244)	1.49	3002	1.00	5092	1.17	207
10(68250)	1.44	6290	1.00	6359	1.18	308
11(68858)	1.72	5891	1.00	4195	1.15	298
12(70009)	2.08	7938	1.00	7160	1.20	346

(表中の PW は Dragon の総配線長に対する比率 (%))

ている。また、最適な配置において各ネットの端子となるセルがすべて近隣に配置される保証はない。表 1 の考察でも述べたが、本手法はネットの端子となるセル同士が近隣に配置されるようなクラスタリングや、概略配置時の評価基準を採用している。しかし、実際の回路では配線長の長いネットも存在するはずであり、このため、配置手法においても一部のネットの配線長が長くなることも考慮する必要がある。本手法のクラスタリングを行う際には配置情報が全くない状態で行っているが、さらに配線長が長くなるネットを前もって考慮することは困難である。しかし、概略配置の際の評価基準の要素に長くなるネットとそうでないネットを区別できるようにして、それに応じた評価値を定めることで配線長が長いネットの存在を許す配置は可能であると考えられる。また、手法の最初に作成したクラスタは概略配置の工程において階層を経るごとに分解していくため、クラスタリングに重点をおくよりはむしろ概略配置を改良することに重点を置くほうが有効であると考えられる。現在は長くなるネットが混在したデータに対して悪い結果が得られたが、実際の回路においても長くなるネットの数は少ないはずであり、上に述べた改良が十分に効果的に働くと Peko データの結果でも分かるようにローカルな最適化は上手くいっているため全体として優れた配置を得ることができると期待している。

5. あとがき

本稿ではクラスタ内の情報を利用してクラスタの配置を行う配置手法を提案した。提案手法ではネットの端子となるセルが最終的な配置において近隣に配置されるようなクラスタリングを行い、階層的に概略配置を求める。提案手法の有効性を示すために行った計算機実験の結果、ローカルな最適化のみで十分なベンチマークデータ (Peko データ) に対しては提案手法は比較手法に比べて優れた結果を得ることが出来た。しかしながら、より実データに近いベンチマークである IBM データに対しては提案手法の改良の必要性を示す結果が得られた。今後は、実データに対しても比較手法より優れた配置結果を得ることを目的として、評価実験から得られた知見に基づき、クラスタリングと概略配置の評価基準を改良する予定である。

文 献

- [1] M.Wang, X.Yang and M.Sarrafzadeh, "Dragon2000 : standard-cell placement tool for large industry circuits," Proc. International Conference on Computer Aided Design, pp. 360-363, 2000.

- [2] A. E. Caldwell, A. B. Kahng and I. L. Markov, "Can recursive bisection produce routable placements?," Proc. Design Automation Conference, pp. 477-482, 2000.
- [3] G. Sigl, K. Doll and F. Johannes, "Analytical placement: A linear or quadratic objective function?," Proc. Design Automation Conference, pp. 427-432, 1991.
- [4] C. Sechen and A. Sangiovanni-Vincentelli, "The Timber-Wolf Placement and Routing Package," Proc. IEEE J of Solid-State Circuits, vol. 20, No. 2, pp. 432-439, 1985.
- [5] X. Yang, B. Choi and M. Sarrafzadeh, "Routability Driven White Space Allocation for Fixed-Die Standard-Cell Placement," Proc. International Symposium on Physical Design, pp. 42-49, 2002.
- [6] Bo Hu and Malgorzata Marek-Sadowska, "Fine Granularity Clustering for Large Scale Placement Problems," Proc. International Symposium on Physical Design, pp. 67-74, 2003.
- [7] "<http://cadlab.cs.ucla.edu/pubbench/peko.htm>"
- [8] "<http://er.cs.ucla.edu/benchmarks/ibm-place2/>"
- [9] "<http://er.cs.ucla.edu/Dragon/download.html>"
- [10] "<http://www.gigascale.org/bk-placement/bin>"