

## 任意形状ブロックのレイアウト手法

中谷 直司 金杉 昭徳<sup>†</sup> 森末 道忠<sup>‡</sup>

<sup>†</sup>埼玉大学工学部

<sup>‡</sup>広島市立大学情報科学部

VLSIレイアウト手法の多くにおいては、配置対象ブロックの形状は矩形に限定され矩形以外のブロックを扱うことは困難である。また矩形以外のブロックを対象とする手法においても、配置対象ブロックの形状自由度は決して高くない。そこで本稿では完全な任意形状ブロックを配置する一手法を提案する。提案する手法はブロックの形状に応じた評価値を定義し、その値を用いることで任意形状ブロックを配置する組立式初期配置手法であり、極めて短時間で結果を得ることが可能であるという特長を有する。また、可能な限り矩形に近い効率の良い配置を得ることができ、任意形状ブロックを対象とした初期配置手法として有用であることを計算機実験により確認した。

## A Layout Method for Arbitrary Shape Blocks

Naoshi NAKAYA Akinori KANASUGI<sup>†</sup> Mititada MORISUE<sup>‡</sup>

<sup>†</sup>Faculty of Engineering Saitama University

<sup>‡</sup>Faculty of Information Sciences Hiroshima City University

In this paper, we propose a new placement method which can treats of arbitrary shape blocks. The proposed method is based on the concept of adjoining degree for blocks and its calculatim time is very short. The computer experiments show that the results of proposed methods are effective placements which are almost rectangle as possible.

## 1 まえがき

多くのVLSIレイアウト手法においては、配置対象ブロックの形状は矩形に限定され矩形以外のブロックを扱う事は困難である。また矩形以外のブロックを対象とする手法においても、そのブロック形状はL型 [1] や凸XY多角形と呼ばれる形状 [2] などに限定され、配置対象ブロックの形状自由度は決して高くない。あるいはブロック形状に制限がない手法 [3] においても、配置領域を限定することは不可能であるという問題がある。そこで本稿ではこれらの問題に対し、形状に応じた評価値を用いることで、任意形状ブロックの効率良い配置を可能とする一手法を提案する。本手法はVLSIレイアウトにおける組立式初期配置手法の1つであり、極めて短い時間で結果を得ることが可能である。但しその一方で、配置処理過程では全体的な見通しが立たないという、すべての組立式手法に共通する欠点をもっているため、本手法の与える結果は必ずしも最良なものではない。しかし、可能な限り矩形に近い効率の良い配置を得ることができ、任意形状ブロックを対象とした初期配置手法として十分に有用であることを計算機実験により確認した。

本手法の任意形状ブロックを扱えるという特長は、タイミングドリブン配置に有効である。すなわち、繰り返し規則性の抽出 [4] されたブロックを1つの任意形状ブロックとして配置することで、タイミング制約にかかわる配線遅延問題に対処することができる。

## 2 提案する手法

本節ではブロック形状に応じた評価値を定義し、その値を用いて任意形状ブロックを配置する一手法を提案する。本手法では配置領域制限がある場合を、配置領域に制限が無いときの特殊な場合としてとらえ配置を行う。したがって、ここでは基本となる配置領域に制限が無い場合の手法から述べることにする。

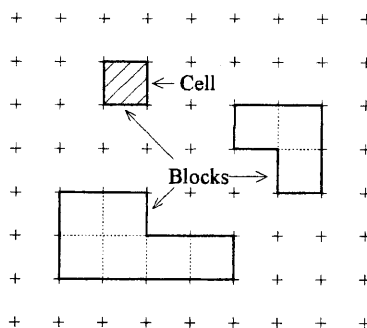


図1 任意形状ブロックの例

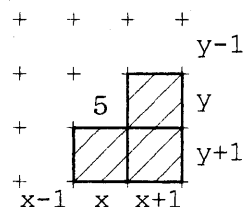


図2 隣接度の例

### 2.1 配置領域に制限が無い場合の手法

配置領域に制限が無い場合の任意形状ブロックの配置について考える。ここでいう任意形状ブロックとは、図1に示すようないくつかのセル（1つのセルは図に斜線で示したもの）の塊である。また、レイアウト領域は基盤の目のように正方形の区画が縦横に連なったものとし、1つのセルは1つの区画を占めるとする。

本手法では、レイアウト領域の各区画に対してブロック形状に応じた評価値を定める。この評価値は、各区画の周りの8つの区画にいくつセルがあるかを重み付きで数えたものであり、その区画にセルを置いたとき既に配置されているセルと隣接する度合を示している。したがって、以下ではこの評価値を「隣接度」とよぶことにする。なお、各セルへの重みは垂直水平方向の4つのセルに対しては2、斜め方向の4つに対しては1とする。すなわち座標  $(x, y)$  の

区画の隣接度  $R_{xy}$  は次式で与えられる.

$$R_{xy} = \sum_{i=-1}^1 \sum_{j=-1}^1 \frac{2}{|i \cdot j| + 1} S_{x+i, y+j} \quad (1)$$

但し  $S_{xy}$ : 座標  $x, y$  の区画に  
セルが存在するとき 1  
" しないとき 0

なお、隣接度は既にセルが配置されている区画については定義しない.

例として図2の場合、中央の区画(座標  $(x, y)$ ) の隣接度は式(1)において、座標  $(x, y+1)$  と  $(x+1, y)$  と  $(x+1, y+1)$  に対応する3つの項が残って

$$R_{xy} = \frac{2}{|0 \cdot 1| + 1} \cdot 1 + \frac{2}{|1 \cdot 0| + 1} \cdot 1 + \frac{2}{|1 \cdot 1| + 1} \cdot 1 = 5$$

となり隣接度は5. 同様にすべての区画の隣接度を求めることができる.

前述したように隣接度はその区画にセルを配置したとき、既に配置されているセルと隣接する割合を示している. したがって、ブロックを配置するときは配置した区画の隣接度の和が最大になる区画に配置すると効率がよく、チップ面積の最少化にもつながる. ここで、ブロック配置の手順を以下に示す.

**ステップ1:** レイアウト領域の各区画に対して式(1)に従い隣接度を求める.

**ステップ2:** ブロックを配置したとき、隣接度の和が最大になる区画を選ぶ. なお、このとき既にセルの存在する区画を選ぶことは許されない.

**ステップ3:** ステップ2で選んだ区画にブロックを配置する.

**ステップ4:** 配置するブロックがなくなるまでステップ1, 2, 3を繰り返す.

例として図3(a)のようにセルが既に配置されているときに、図3(b)のブロックを配置する場合を考える. ステップ1に従い隣接度を求めると、その分布は図3(c)のようになる(図に示した以外の区画の隣接度は0である).

この隣接度分布から考えてステップ2で選ばれた区画は、図3(d)の斜線の部分になる. したがって、ステップ3で配置した結果は図3(e)となる. ここでまだ配置するブロックが残っている場合は、図3(e)に示したセル配置に対し隣接度分布を求め(つまり、ステップ1に戻る)、同様に繰り返す.

本手法の任意形状ブロックを扱えるという特長は、タイミングドリブンの配置に有効である. ある回路において繰り返し規則性の抽出されたブロックでは、ブロック内の配線長を同じにしたタイミングのずれを回避することが望ましい. そこでブロックの形状、構成がまったく同じならばその中に含まれる配線長、配線遅延も同じになりタイミングのずれは生じ得ないことに着目し、繰り返し規則性のあるブロックを同形状、同構成のブロックとして扱うという対処方法が考えられる. このとき繰り返し規則性の抽出されたブロックが素数個で構成されていると、矩形のブロックを得ることは事実上できない(一直線に並べることはできる). あるいはブロック内での配線長を短くするため、あえて矩形以外の形状にしたいことも考えられる. こういった場合に任意形状ブロックを配置する必要が生じ、本手法は有効であるといえる.

## 2.2 配置領域に制限がある場合の手法

次に、配置領域に制限がある場合について考える. 既に述べた配置領域に制限が無い場合においては、レイアウト領域は無限の広がりを持つものとして扱ってきた. そこで配置領域に制限がある場合においても同様に、レイアウト領域は無限の広がりを持つものとする. そして配置領域制限は、レイアウト領域の配置可能区画以外はすべてセル(このセルを以下では「配置領域制限のセル」とよぶ)で埋め尽くされている、とすることで実現する. これにより配置領域に制限がある場合においても同様に、隣接度を求めその和が最大になる区画に配置するという手法を使うことができる. なお、この場合「配置領域制限のセル」よりも「配置対象ブロックのセル」により隣接した区画へ配置を行うために、この両者に対する重みを変え隣接度を式

(1) を拡張した次式で定義する.

$$R_{xy} = \sum_{i=-1}^1 \sum_{j=-1}^1 \frac{2}{|i \cdot j| + 1} S_{x+i, y+j} + S'_{x+i, y+j} \quad (2)$$

但し  $S_{xy}$  : 座標  $x, y$  の区画に  
配置対象ブロックのセルが  
存在するとき 1  
" しないとき 0

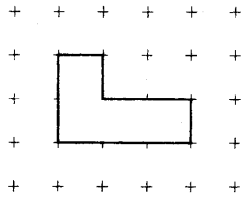
$S'_{xy}$  : 座標  $x, y$  の区画に  
配置領域制限のセルが  
存在するとき 1  
" しないとき 0

なお、隣接度は既にセルのある区画については定義しない。

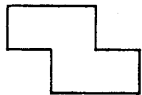
一例として4行5列の配置領域制限がある場合、レイアウト領域は図4(a)のように4×5の配置可能区画を残してセルで埋め尽くされているとして考える。これに対し隣接度を式(2)に従い求めると、図4(b)のような分布が得られる。後は配置領域に制限が無い場合と同様に、既に述べたブロック配置の手順に従いブロックを配置する。但し、このとき隣接度は式(2)で定義されたものを用いる。

なお、この例では配置領域制限を矩形としたが、本手法としては領域制限が矩形である必要はない。たとえ領域制限が非矩形であっても、隣接度を求める上で問題はないため配置は可能である。

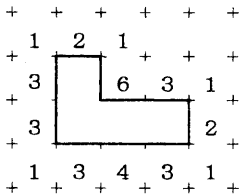
また、配置領域制限内に配置対象のブロックが収まるためにはいくつかの条件がある。1つは領域制限の面積がブロックの総面積よりも大きいこと、言い換えるならば配置可能区画の数が、ブロックの総セル数よりも多いこと。2つめは各ブロックを内包する矩形を考えたとき、それら矩形の長辺の中で最も長いものが領域制限(この場合矩形で考えている)の長辺の長さを越えないこと。この2つの条件を満たすことは最低限必要であり、さらに次の条件を満たさなければならない。すなわち、ブロック全体の構成(どういった形状のブロックがいくつかあるか)が、領域制限内に収まり得る構成であること。だが、この「収まり得る構成である」こと



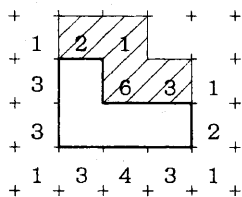
(a) 配置済みブロック



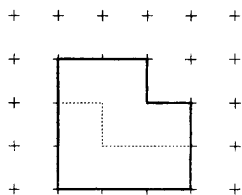
(b) 次に配置するブロック



(c) 隣接度の分布

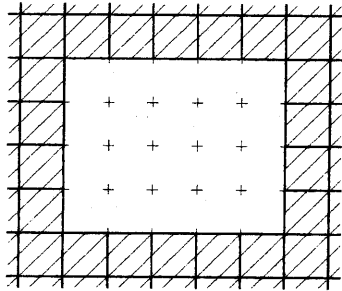


(d) 隣接度の和が最大になる区画

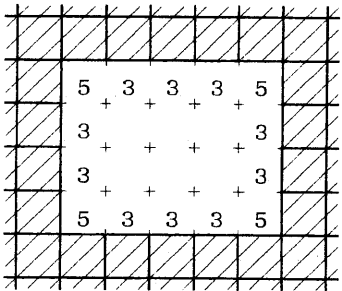


(e) 配置結果

図3 隣接度による配置の例



(a) 配置領域制限の例



(b) 配置領域制限の隣接度の例

図 4 領域制限があるときのレイアウト領域

を完全に証明する方法はない。与えられた構成が収まらないことはパリティを使った方法で証明可能 [5] だが、収まらなると証明されなかったとしても収まるとは限らない。あくまでも収まる可能性があるだけで実際に収めてみる以外に確認する方法はない。この3つめの条件は領域制限を設けるときに問題ではあるが、収まらない構成を避ける以外に具体的な対処法はないのが現実である。

### 3 計算機実験

本手法の有用性を確認するため計算機実験を行った。本手法においては配置結果は配置順序に依存するため、配置順序には適当な制御を要する。本実験では接続関係の大小を基本として、配置順序を決定した。まず、ブロック間の接続関係を示す接続行列を次のように定義する。

対象とするモデルが  $n$  個のブロック  $b_i$  ( $i = 1, \dots, n$ ) から成るとき、接続行列  $C$  の各要素

$c_{ij}$  ( $i, j = 1, \dots, n$ ) はブロック  $b_i$  と  $b_j$  間の接続本数を示す。但し対角要素  $c_{ii}$  はブロック  $b_i$  に接続する総本数  $\sum_{j \neq i} c_{ij}$  とする。

この接続行列を用い、以下の手順で配置順序を決定した。

**ステップ1:** 接続行列  $C$  から  $\max c_{ii}$  を満たす  $i$  を求め、ブロック  $b_i$  を配置する。

**ステップ2. 1:** 次に  $\max c_{ij}$  を満たす  $j$  を求めステップ3へ。もし  $j$  が1つに特定できないときはステップ2. 2へ。

**ステップ2. 2:** ステップ2. 1で選ばれた  $j$  の中から、さらに  $\max c_{jj}$  を満たす  $j$  を求めステップ3へ。もし  $j$  が1つに特定できないときはステップ2. 3へ。

**ステップ2. 3:** ステップ2. 2で選ばれた  $j$  の中から、さらにブロック  $b_j$  の面積が最も大きい  $j$  を選びステップ3へ。もし  $j$  が1つに特定できないときは、それらの  $j$  は等価と考へ一番小さい  $j$  を選びステップ3へ。

**ステップ3:** ブロック  $b_j$  を配置し、ブロック  $b_i$  と  $b_j$  を合成したものを新たなブロック  $b_i$  とし、接続行列  $C$  を求め直す。

**ステップ4:** 配置するブロックがなくなるまでステップ2, 3を繰り返す。

したがって、例えば図5に示したブロックを配置するとき、この手順により配置順序はブロック番号7, 6, 8, 3, 2, 5, 4, 1の順となる。また、隣接度は形状に応じた評価値であるため対称形では隣接度も対称となり、ブロックを配置したときに隣接度の和が最大になる区画も複数になる。そこで配置区画候補が複数になった時の選択法として他の評価法との併用が必要となる。本実験ではブロック中心間のマンハッタン距離による仮想配線長を用いた。なお、ブロック中心はそのブロックを内包する最少の矩形の重心と定義した。

例として図5に示したブロックを、配置領域制限を設けずに配置する。図5のブロックでは配置途中で隣接度による評価が同じで、なおか

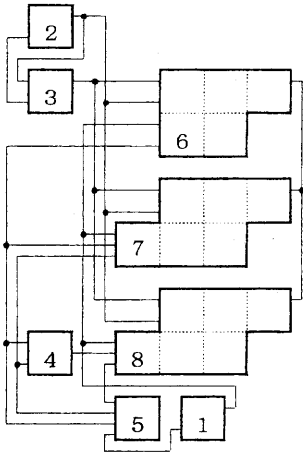


図 5 配置対象ブロックの例 1

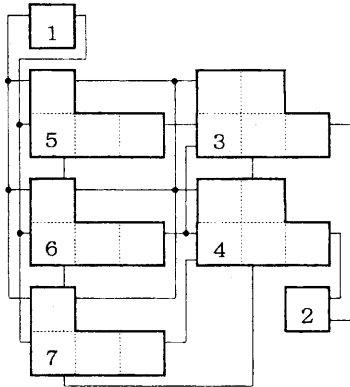


図 6 配置対象ブロックの例 2

つ仮想配線長による評価も同じになる配置区画候補が複数存在し、配置結果が9つに分岐する。そのうち最も仮想配線長の短い(セルの1辺を1として110.5)配置結果を図7に示すが、他の8つ配置結果も図7と同様に矩形に近い配置が得られた。

また、同様に図6に示すブロックを配置すると、配置順序はブロック番号6, 7, 5, 4, 3, 1, 2の順となり8つの配置結果が得られる。そのうち最も仮想配線長の短い(93)配置結果を図8に示す。この場合も他の7つは図8と全く同じ外形の配置が得られた。

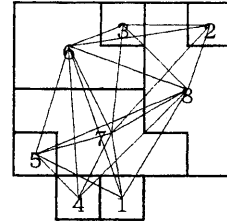


図 7 領域制限が無いときの配置 (例 1)

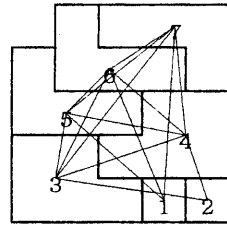


図 8 領域制限が無いときの配置 (例 2)

配置領域に制限がある場合についても計算機実験を行った。ここでも例として図5に示したブロックを配置することを考える。図5に示したブロックの総セル数は22、よって4行6列の配置領域制限を設けることとした。また、それ以外の条件(配置順序、および仮想配線長の併用)については配置領域制限が無い場合と同様とした。結果は図9に示すように、与えられた配置領域内にすべてのブロックが配置された。なお、仮想配線長は117.5で配置結果の分岐は起こらなかった。

また、同様に図6に示すブロックを4×6の配置領域に配置した。配置結果は3つに分岐するが、3つとも与えられた配置領域内にすべてのブロックが配置された。そのうち最も仮想配線長の短い(97)配置結果を図10に示す。

本手法においては、配置領域制限は必ずしも矩形である必要はない。そこで図6に示すブロックを図11のような非矩形の配置領域に配置してみた。結果は図12に示すように与えられた配置領域内にすべてのブロックが配置された。なお、この例では配置結果が2つに分岐するため、仮想配線長の短い(99)ものを示した。

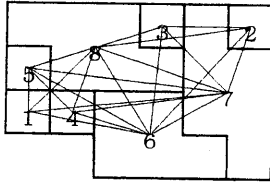


図 9 領域制限があるときの配置 (例 1)

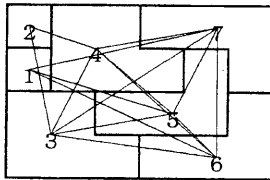


図 10 領域制限があるときの配置 (例 2)

#### 4 むすび

本稿ではVLSIレイアウトにおける組立式初期配置手法の1つとして、ブロック形状に応じた評価値を定義することで、任意形状ブロックを効率良く配置する一手法を提案した。本手法は極めて短時間で、かつ可能な限り矩形に近い効率良い配置結果を与えるという特長を有する。また、配置領域に制限がある場合を制限が無い場合の特殊な場合としてとらえ、全く同じ手法で結果を得ることが可能である。

今後の課題として、最終的配置結果がブロックの配置順序に強く依存するため、本手法に適した配置順序の検討が必要である。

#### 参考文献

- [1] Wong D.F. and Liu C.L. : "Floorplan Design for Rectangular and L-shaped Modules" ,Proc. Intl. Conf. on Computer-Aided-Design,pp.520-523,1987.
- [2] 大村道郎, 宮尾淳一, 菊野 亨, 吉田典可 : "VLSIのブロック配置における重なり除去問題", 信学論 (A), Vol.J72-A, No.7, pp.1093-1100, Jul. 1989.
- [3] Hudson J.A.,Wisniewski J.A. and Peters

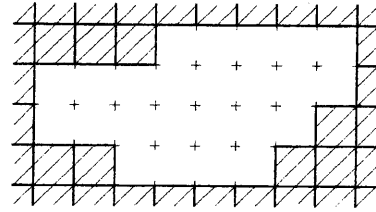


図 11 非矩形の領域制限

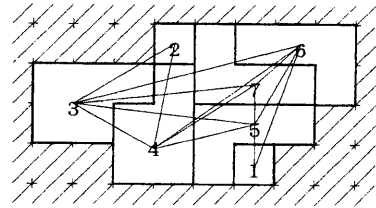


図 12 非矩形の領域制限があるときの配置 (例 2)

R.C. : "Module Positioning Algorithms for Rectilinear Macrocell Assemblies" ,21st Design Automation Conference,pp.672-675,1984.

- [4] 金杉昭徳, 佐藤修司, 森末道忠 : "論理回路における規則性抽出の一手法", 信学技報, VLD91-69, Sep. 1991.
- [5] J. D. ビースリー : "ゲームと競技の数学", サイエンス社, 1992.
- [6] 渡辺孝博 : "LSIレイアウト自動設計の現状と可能性", 信学誌, Vol.76, No.7, pp.774-782, Jul. 1993.
- [7] 小林 勉, 須藤常太, 細田泰弘 : "LSI-CAD [1]", 信学誌, Vol.70, No.12, pp.1291-1297, Nov. 1987.
- [8] 上田和宏, 須藤常太 : "LSI-CAD [2]", 信学誌, Vol.71, No.1, pp.80-87, Jan. 1988.
- [9] 佐藤真司, 堤 定雄, 後藤源助 : "マスタスライスセルの自動発生", 信学技報, ICD88-70, 1988.

- [10] 中谷直司, 金杉昭徳, 森末道忠: “任意形状  
ブロックの一配置手法”, プリント回路学会  
第8回学術講演大会講演論文集, pp. 85-86,  
Mar. 1994.