

アナログIC設計における 排他的近接対称配置制約を考慮した配置手法の研究

浅野 晋平[†] 藤吉 邦洋[†]

[†] 東京農工大学大学院 電気電子工学専攻

あらまし アナログICのレイアウト設計では、しばしば複数のセル対を水平、もしくは垂直な軸に対して線対称に配置することが要求される。そこで、矩形配置表現方法である sequence-pair を用いて、その表現による制約と、対称配置制約とを満たした配置を得る手法が提案されてきた。対称に配置すべきセル対は、出来るだけ近くに配置する事も同時に求められる場合が多く、このことから近接対称配置制約が提案された。しかし、この制約だけでは対称に配置したセルの組同士が離れ、配線による寄生素子の特性に不整合が生じる可能性がある。そこで本稿では、対称に配置するだけでなく、ある対称軸に対称に配置せよとされたセル同士を出来るだけ近くに配置する制約として、排他的近接対称配置制約を定義する。そして、凸多角形パッキングの手法を応用することで、制約に基づく配置をより高速に得ることが出来る手法を提案する。そして実験によって提案手法の有効性を確認する。

キーワード 対称配置制約, sequence-pair, アナログIC, 配置

On Handling Cell Placement with Exclusive Adjacent Symmetry Constraints for Analog IC Layout Design

Shimpei ASANO[†] and Kunihiro FUJIYOSHI[†]

[†] Department of Electrical and Electronic Engineering, Tokyo University of Agriculture and Technology

Abstract In recent high performance analog IC design, it is often required to place some cells symmetrically to a horizontal or vertical axis. Then, some methods of obtaining the closest placement that satisfies the given symmetry constraints and the topology constraints imposed by a sequence-pair were proposed. But, some cells placed symmetrically are required to be placed nearby. Therefore, in this paper, we define “exclusive adjacent symmetry constraint” and propose a method of obtaining the closest cell placement that satisfies the given constraints.

Key words symmetry constraint, sequence-pair, analog IC, placement

1. ま え が き

アナログICのレイアウト設計において、しばしばトランジスタや抵抗等を指すセル対がいくつか指定され、これらを線対称に配置配線する[1]。これは、セルやそれと接続する配線のレイアウト、及びそれによって生ずる寄生素子の整合をとることで、オフセット電圧が高くなることや電源電圧変動除去比(PSRR)の低下を避けるためである[2]。このように、指定されたセル対を線対称に配置しなければならないという制約を“対称配置制約”と呼ぶ。対称軸は一つとは限らず、一つの対称軸に対して対称に配置しなければならないセルの集合を対称集合という。

これまでアナログIC設計では人手による設計が主流であったが、近年では矩形配置の表現法と Simulated Annealing 法(SA法)とを組み合わせた自動配置手法に対して対称配置制約を課し、準最適な配置を得る手法が提案されてきた。Balasaらは、対称配置制約と、セル配置をセル名の順列の対で表現する

手法である sequence-pair が示唆する制約とを満たす最密なセル配置を得るという手法を提案した[1]。そしてこれをSA法に組み込むことで、対称配置制約を満たして密な配置を自動で高速に求めることを可能とした。

しかし、Balasaらの手法では対称配置制約しか考慮していないため、対称に配置されたセル同士が離れてしまう場合がある。セルを対称に配置するのはセル特性の整合をとるためであるが、対称に配置されていてもセル間が離れてしまうとその特性に不整合が生じてしまう。

そこで甲田らは、同じ対称集合に属する対称セル対間にその対称集合に属さないセルが入り込むことを禁じ、対称セル対を対称軸に対して対称かつ出来るだけ近接して配置するという“近接対称配置制約”を定義し、この制約と sequence-pair が示唆する制約を満たした配置の中で最密なものをセル数 n に対して $O(n^3)$ 時間で得るデコードアルゴリズムを提案した[3]。これにより、対称集合に属するセルが対称軸に近接して配置され、

セル内の素子特性の整合が取れた配置を得られるようになった。しかし、この手法では同じ対称集合に属するセル同士が対称軸と平行の方向に離れてしまう可能性があり、すると配線による寄生素子の整合が取れなくなってしまう。

そこで本研究では近接対称配置制約を強め、同じ対称集合に属する任意のセルの間にその対称集合に属さないセルが入り込むことを禁じ、同じ対称集合内のセルが出来るだけ近付くよう制約した「排他的近接対称配置制約」を定義する。そしてこの制約と sequence-pair が示唆する制約を満たす配置を求めることを、凸多角形パッキングの手法 [7] を用い $O(n^2)$ 時間で行うことのできるデコードアルゴリズムを提案する。また、計算機実験結果により提案手法の有用性を検証する。

2. パッキング手法

2.1 Sequence-Pair [4]

sequence-pair (以下 seq-pair) では、 n 個の矩形の相対的な位置関係を矩形名の順列 Γ_+ と Γ_- の対により、 (Γ_+, Γ_-) の形で表す。当然、 n 個の矩形の配置に対して $(n!)^2$ 通りの表現がある。ここで、 $\Gamma_+(i)$ は Γ_+ 中で第 i 番目の矩形のことを指し、 $\Gamma_+^{-1}(a)$ は Γ_+ 中で矩形 (a) が左から何番目かを指す。 Γ_- についても同様である。

seq-pair では矩形対の相対的な位置関係を、以下に示す「水平/垂直制約」として表す。 Γ_+ と Γ_- で共に a が b の前にあるとき、つまり $\Gamma_+^{-1}(a) < \Gamma_+^{-1}(b)$ かつ $\Gamma_-^{-1}(a) < \Gamma_-^{-1}(b)$ であるとき、矩形 a は矩形 b の左に位置する。また Γ_+ では a が b の前にあり、 Γ_- では a が b の後ろにあるとき、すなわち $\Gamma_+^{-1}(a) < \Gamma_-^{-1}(b)$ かつ $\Gamma_-^{-1}(a) > \Gamma_-^{-1}(b)$ であるとき、矩形 a は矩形 b の上に位置する。例えば、seq-pair $(abcd; bdac)$ は、図 1 のような相対位置関係を表す。

seq-pair が示唆する水平/垂直制約は水平と垂直の制約グラフという枝に重みのついた有向無閉路グラフによって表すことができる。具体的には水平制約グラフは各矩形に点に対応し、水平制約関係にある全ての矩形対について左側の矩形に対応する点から右側の矩形に対応する点に有向枝が張られ、出点に対応する矩形の横幅がその枝の重みになったものに大 source 点を付加し、そこから各点へ重み 0 の枝を張ったものである。垂直制約グラフは枝が垂直制約関係により張られ、重みが矩形の高さであること以外は同じである。

水平/垂直制約グラフに対して、よく知られた DAG の最長パスアルゴリズムを用いて大 source 点から各点への最長パス長を求めて、これに対応する矩形の左下座標とすることで、seq-pair の示唆する制約の下で最も密である左下詰めパッキングを $O(n^2)$ 時間で得ることができる。

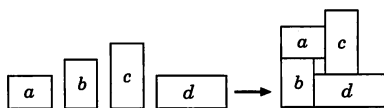


図 1 seq-pair $(abcd; bdac)$ に基づいた左下詰め配置

2.2 seq-pair に基づいてレクトリニア多角形パッキングを求める手法 [5]

外周が水平線分と垂直線分のみで構成されている多角形のことをレクトリニア多角形 (以下、多角形) という。[5] において seq-pair を拡張することで、レクトリニア多角形パッキン

グを求める手法が提案された。seq-pair は矩形のみしか扱うことができないため、ここで各多角形は水平/垂直線分で切断された部分矩形の集合として扱われている。そして、与えられた seq-pair に基づいた左下詰めパッキングを $O(n^3)$ 時間で求めることが出来る手法が示された。ただし n は部分矩形とセルの個数である。

2.3 凸多角形のパッキング手法 [7]

外周のどの 2 点をとっても、それらを結ぶ最短マンハッタンパスの少なくとも一つが内包されるレクトリニア多角形を「凸多角形」と呼び、そうでない多角形を「凹多角形」と呼ぶ。対象を凸多角形に限定することにより、 $O(n^2)$ 時間で与えられた seq-pair に基づいた左下詰めパッキングを求める手法が提案された [7]。この手法は、凸多角形からできた部分矩形が seq-pair 上で元の凸多角形を復元できるように並んでなければいけないという correlative 制約と以下に示す separate 制約を満たした seq-pair だけを対象としているが、これらの制約を満たした seq-pair はどんな凸多角形パッキングも表せる。また、これらの制約を満たした seq-pair の示唆する上下左右制約を守った上で全ての凸多角形を復元したパッキングを必ず求めることが出来る。

2.3.1 separate 制約

凸多角形 a と b について、 Γ_+ において、 a の部分矩形の集合の要素全てが b の部分矩形の集合の要素全ての前、または後ろにあるとき、「 a と b は Γ_+ において separate である」という。 Γ_- においても同様である。

separate 制約: 全ての凸多角形対は必ず、 Γ_+ か Γ_- で separate でなくてはならない。ここでの凸多角形は矩形も含む。

例えば元は一つの凸多角形 a と b と c であった部分矩形の集合 $\{a_1, a_2\}, \{b_1, b_2\}, \{c_1, c_2\}$ の seq-pair が $(a_1 c_1 a_2 b_1 b_2 c_2; a_1 c_1 c_2 b_1 b_2 a_2)$ であった場合、 a と b は Γ_+ で、 b と c は Γ_- で separate であるが、 a と c が Γ_+, Γ_- 両方で separate でない。従ってこの seq-pair は separate 制約を満たしていない。

2.3.2 凸多角形パッキングのデコードアルゴリズム

入力: 各矩形のサイズと、各部分矩形集合を元の凸多角形に復元するのに必要な座標情報、correlative 制約と separate 制約を満たした seq-pair。

step1: 水平制約グラフ G_H^S の作成

凸多角形を点とし、seq-pair において凸多角形 a の部分矩形 a_i が凸多角形 b の部分矩形 b_j の左に制約されると、 a から b へ重み

$$\max_{i,j} (a_i \text{の幅} + d_x(a_i) - d_x(b_j) \mid a_i \text{は } b_j \text{の左に制約}) \quad (1)$$

の枝を張る。ここで $d_x(a_i)$ は凸多角形 a の最左 x 座標と a_i の左辺 x 座標の差である。

そして大ソース点を付加し、そこから他の全ての点へ重み 0 の枝を張る。その後、元は同じ凸多角形であった部分矩形の集合を一つの点に融合する (超点化)。

step2: 最長パス計算による多角形の x 座標決め

G_H^S において大ソース点から各点への最長パス長を DAG 用の最長パスアルゴリズムを用いて求め、対応する凸多角形の最左 x 座標とする。

step3: 垂直方向についての計算

step1, step2 と同様にして垂直制約グラフ G_V^S を作成し、最下 y 座標を決める。 ■

例えば、図 2(a) の多角形と seq-pair $(ca_1 a_2 b_1 b_2; a_1 b_1$

$a_2 c b_2$ が入力されたとする。step1 に得られる水平制約グラフを図 2(b) に示し、超点化を行ったグラフを図 2(c) に示し、step3 で得られる超点化後の垂直制約グラフを図 2(d) に示す。最終的に得られる左下詰め配置を図 2(e) に示す。

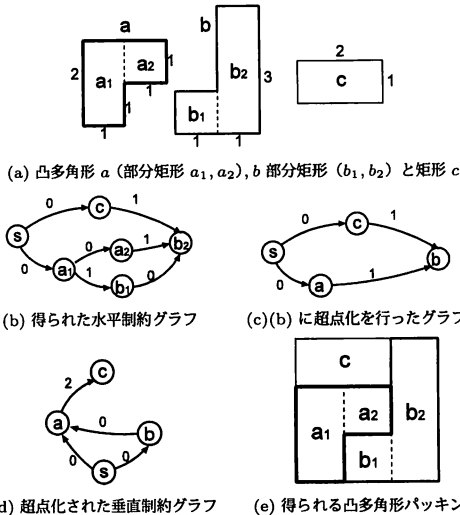


図 2 seq-pair $(c a_1 a_2 b_1 b_2; a_1 b_1 a_2 c b_2)$ と凸多角形 a (部分矩形 a_1, a_2), b (部分矩形 b_1, b_2) と矩形 c から得られるパッキング。簡単のため推移枝は省略

ここで得られるパッキングは、左下詰めであることから分かるように最密である。この手法は得られるグラフが DAG であることが示されているため、DAG の最長パスアルゴリズムを使うことで、セル数が n のとき $O(n^2)$ 時間で実行可能である。

3. 対称配置

3.1 対称配置制約 [1]

対称配置制約は、複数のセル対からなる対称集合について、その中の各セル対を共通の垂直または水平の対称軸に対して線対称になるように配置せよという制約である。対称集合の中にはそれぞれひとつで対称に配置せよという自己対称セルが含まれる場合もある。これはつまり対称軸の x または y 座標にその中心を合わせなければならないセルである。対称集合は 1 つとは限らず、各々の対称集合ごとに 1 本の垂直または水平な軸に対して対称にその集合のセル対を配置する。

垂直な対称軸に対して対称に配置するセル対の左のセル、右のセルをそれぞれ左セル、右セルと呼び、 a_l, a_r と矩形名に添字 l, r をつけて表す。水平な対称軸に対しては、対称に配置するセル対の上のセル、下のセルをそれぞれ上セル、下セルと呼び、 a_u, a_b と矩形名に添字 u, b をつけて表す。自己対称セルは a_s の様に添字 s をつけて表す。対称集合は対称に配置するセル対を括弧でまとめ $\{(a_l, a_r), b_s\}$ の様に表記する。そして、対称集合を列挙することで対称配置制約を表現する。

3.2 近接対称配置制約 [3]

アナログ IC 設計において、対称配置制約はセル同士の特性の整合を取るために与えられているが、対称配置制約だけでは配置される位置が離れてしまい、セル特性にばらつきが出て整合が取れなくなる場合がある。例えば seq-pair $(a_l b_l a_r c b_r d; c b_l b_r d a_l a_r)$ と対称集合 $\{(a_l, a_r), (b_l, b_r)\}$

が与えられた場合、対称配置制約を満たした左下詰め配置は図 3(a) のようにセルが対称に配置されてもその間が空いてしまう。

そこで、対称軸に出来るだけ近接してセルを配置させるという近接対称配置制約が定義された [3]。これは、以下の 3 つの制約からなる。

(制約 1) 対称集合毎に、対称セル対を対称軸に対して対称に配置する。

(制約 2) 対称集合毎に、対称セル対を出来るだけ対称軸に近づけて配置する。

(制約 3) 同じ対称集合に属する任意の左セルと右セルの間、もしくは下セルと上セルの間に、その対称集合に属さないセルが入り込むことを禁じる。

先ほどの例の seq-pair と対称集合に近接対称配置制約を課した場合に得られる配置では、図 3(b) に示されているように対称集合に属するセルが軸に対して近接に配置されたものとなる。

なお、セル対を対称軸に近接させることにより図 3(b) のように対称軸に対して垂直な方向の幅が大きくなってしまふ場合があるが、アナログ IC 設計においては、より小さな面積で配置することよりも、回路性能をあげるために与えられた制約を満たす方が重要であるため、本稿ではこのような面積の増大を許容する。

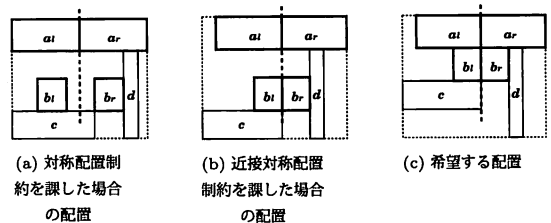


図 3 seq-pair $(a_l b_l a_r c b_r d; c b_l b_r d a_l a_r)$ と対称集合 $\{(a_l, a_r), (b_l, b_r)\}$ から得られる左下詰め配置

4. 排他的近接対称配置制約と提案するデコードアルゴリズム

対称に配置するセルを近接対称配置制約により対称軸に近接させて対称に配置することで素子特性の整合を取り易くなるが、配線による寄生素子の特性の整合を取るには図 3(c) のように軸と平行な方向に対しても対称集合セルが近接することが望ましい。

そこで対称集合内のセルが不必要に離れてしまうことを禁じ、対称軸と平行な方向に対しても詰めた配置を得るため、3.2 に示した近接対称配置制約の (制約 2) に「対称軸に平行な方向に関して最小の幅内に配置する」という制約を加える。また、対称集合に属さないセルにより最小の幅内への配置を妨げられないために (制約 3) をさらに厳しい「同じ対称集合に属する任意のセルの間に、その対称集合に属さないセルが入り込むことを禁じる」という制約にする。これらにより得られる以下の制約を排他的近接対称配置制約と定義する。

(制約 1) 対称集合毎に、対称セル対を対称軸に対して対称に配置する。

(制約 2') 対称集合毎に、対称セル対を出来るだけ対称軸に近づけて配置する。また、対称軸に平行な方向に関して最小の幅内に配置する。

(制約 3') 同じ対称集合に属する任意のセルの間に、その対称集合に属さないセルが入り込むことを禁じる。

以降では、対称集合全てにこの排他的近接対称配置制約が課されているものとする。なお、排他的近接対称配置制約も対称集合と同じ表記法で表現する。この制約により対称軸と平行な方向の幅が増大することがあるが、3.2と同様に許容する。

4.1 (制約 3') と separate 制約の関係

排他的近接対称制約が $\{(a_l, a_r), (b_l, b_r)\}$ で与えられたとき、seq-pair $(a_l c b_l a_r b_r; b_l c a_l b_r a_r)$ に基づいた配置は図 4 に示す様に、対称集合に属すセル a_l と b_l の間に対称集合に属さないセル c が入り込んでおり、前述の(制約 3')を満たしていない。この例から分かるように、(制約 3')は seq-pair においては、「 Γ_+ と Γ_- の両方において、対称集合に属さないセルが対称集合に属するセル対の間に挟まれてはならない」と読み替えることが出来る。

すると、前章で説明した separate 制約を満たしていればこの制約を満たしていることは、この制約を満たしていない $(\dots a_l \dots c \dots b_l \dots; \dots b_l \dots c \dots a_l \dots)$ の様な seq-pair が明らかに separate 制約を満たしていないことから容易に分かる。そこで以降では、(制約 3')に替えて separate 制約を用いることにする。

但し、(制約 3') と separate 制約は等しくはない。これは例えば、対称集合二つ $\{(a_l, a_r), \{(b_l, b_r)\}$ で排他的近接対称制約が与えられたとして、seq-pair が $(a_l b_l b_r a_r; b_l a_l a_r b_r)$ だと考えると、この seq-pair は明らかに(制約 3')は満たしているが separate 制約は満たしていない。

このように separate 制約の方が厳しい制約ではあるが後述の定理 1 に示すように、この制約を課しても seq-pair は排他的近接対称配置制約を満たしたどんなパッキングでも表現することができる。

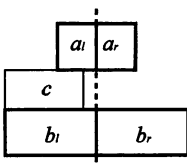


図 4 対称集合内のセル間に対称集合に属さないセルが入り込んでしまう配置例

4.2 symmetric feasible sequence-pair

Balasa らは [1] において、それぞれの対称集合について次式 (2) を満たす seq-pair を、symmetric feasible sequence-pair と定義した。なお、ここではセル a と対称に配置すべきセルを $sym(a)$ と表している。

$$\Gamma_+^{-1}(x) < \Gamma_+^{-1}(y) \Leftrightarrow \Gamma_-^{-1}(sym(y)) < \Gamma_-^{-1}(sym(x)) \quad (2)$$

対称軸が垂直線である場合、式 (2) は

$$\Gamma_+^{-1}(x_l) < \Gamma_+^{-1}(y_l) \Leftrightarrow \Gamma_-^{-1}(y_r) < \Gamma_-^{-1}(x_r)$$

となる。

Balasa らは対称集合が一つだけなら symmetric feasible

sequence-pair には必ず制約を満たしたパッキングが存在することを示した。また、対称配置制約を満たした配置は必ず symmetric feasible sequence-pair によって表現可能である、と主張していた。

しかし、対称配置制約を満たしてはいるものの symmetric feasible sequence-pair では表現できない配置が存在する。例えば対称集合 $\{(a_l, a_r), (b_l, b_r)\}$ で表される制約を満たす図 5 に示した配置に対応する seq-pair は $(b_l b_r c a_l d a_r; a_l c b_l b_r d a_r)$ だけだが、これは symmetric feasible sequence-pair ではない。

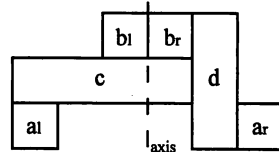


図 5 対称配置制約 $\{(a_l, a_r), (b_l, b_r)\}$ は満たしているが、対応する唯一の seq-pair $(b_l b_r c a_l d a_r; a_l c b_l b_r d a_r)$ が symmetric feasible でない配置

しかし排他的近接対称配置制約が与えられている場合、以下の定理を満たす。

【定理 1】 排他的近接対称配置制約を満たした配置は必ず、separate 制約を満たした symmetric feasible sequence-pair で表現可能である。

(証明) 排他的近接対称配置制約を満たした任意の配置に対し、対称集合毎にそれを囲む最小の凸多角形を考慮して、[7] で提案されている separatable gridding を用いれば separate 制約を満たした seq-pair が必ず得られることは [7] の定理 3 から容易に分かる。また、gridding の際に左右と上下方向のどちらかを優先すると symmetric feasible sequence-pair になる。■

4.3 提案するデコードアルゴリズム

排他的近接対称配置制約と symmetric feasible sequence-pair が示唆する制約を共に満たす配置を、凸多角形パッキング手法 [7] を利用することで $O(n^2)$ 時間で求める手法を提案する。提案するアルゴリズムは 1st step と 2nd step から成る。

1st step では separate 制約を満たした symmetric-feasible seq-pair を対称集合毎に近接させるために、甲田らが近接対称配置制約のために [3] において提案したアルゴリズムを用い、対称集合毎にそれぞれのセルの相対位置を出来るだけ近接させた位置に決定する。

2nd step では 1st step で求めた対称集合内での相対座標を凸多角形を復元する座標情報として、2.2 で述べた手法を使い各セルの左下座標を求める。ここで求めた凸多角形の左下座標と 1st step で求めた対称集合内での相対座標により各セルの左下座標が求まる。

対称軸が垂直線であった場合について 1st step を詳しく述べると、対称集合ごとのセルの相対座標の決定は以下のように行う。対称軸が水平軸であった場合も同様にして相対座標を求めることが出来る。

a) 水平制約グラフを用いた相対 x 座標の決定

seq-pair から水平制約グラフ G_H を作る。以下の作業は対称集合毎に行う。

注目している対称集合について右セルと自己対称セルに対応する節点集合によってできる誘導部分グラフを求め G'_H とする。 G'_H に source 点を加え、この点から各点へ枝を張る。このとき入点に対応するセルが自己対称セルならば枝重みは自己対称セル幅 w_s に対して $-w_s/2$ とし、右セルに対応する点ならば枝重みは 0 とする。

source 点から各点への最長パス長を求め、それを各点に対応するセルの左辺の相対 x 座標とする。

右セルと対になる左セルの左辺の相対 x 座標を、左セルの幅 w_l に対して $-(w_l + (\text{右セルの対称軸との相対 } x \text{ 距離}))$ から求め、その対称集合内で最小の相対 x 座標を持つセルの左辺の相対 x 座標を基準として各セルの相対 x 座標を決定する。

b) 垂直制約グラフを用いた相対 y 座標の決定

seq-pair から水平制約グラフ G_H を作る。以下の作業は対称集合毎に行う。

水平制約グラフと同様に右セルと自己対称セルに対応する点によって定まる誘導部分グラフ G'_V を作り、source 点を加えそこから各点へ重み 0 の枝を張る。source 点から各点への最長パス長を求め、それを対応するセルの下辺の相対 y 座標として決定する。

右セルと対になる左セルの下辺の相対 y 座標は右セルと同値とする。

例えば、全ての矩形のサイズが 2×2 の正方形であると、排他的近接対称配置制約 $\{(a_l, a_r), b_s\}$ と、 $\{(d_u, d_b), e_s\}$ と、symmetric-feasible seq-pair $(b_s, a_l, a_r, c, d_u, d_b, e_s; a_l, a_r, b_s, c, d_u, d_b, e_s)$ が入力された場合。入力された seq-pair から得られる水平/垂直グラフは図 6(a),(b) になる。1st step から得られる、対称集合 $\{(a_l, a_r), b_s\}$ についての G'_H, G'_V を図 6(c),(d) に、対称集合 $\{(d_u, d_b), e_s\}$ についての G'_H, G'_V を図 6(e),(f) に示す。

これより $d_x(a_l) = 0, d_x(a_r) = 2, d_x(b_s) = 1, d_x(d_u) = 0, d_x(d_b) = 0, d_x(e_s) = 2, d_y(a_l) = 0, d_y(a_r) = 0, d_y(b_s) = 2, d_y(d_u) = 2, d_y(d_b) = 0, d_y(e_s) = 1$ となり step2 により G_H として図 6(g) を得て、超点化により図 6(i) を得る。同様に、 G_V は図 6(h) を得て、超点化により図 6(k) を得る。

以上の結果より得られる配置は図 6(l) のようになる。

【定理 2】 separate 制約を満たした symmetric feasible sequence-pair を入力としたとき、提案したアルゴリズムは制約を満たした配置を $O(n^2)$ 時間で得ることが出来る。

5. SA 法探索による配置実験

提案手法の有効性を確認するために、提案手法を以下のように SA 法に組み込み計算機実装した。

5.1 隣接解生成手法

ランダムにセルを 2 つ選び Γ_+ 及び Γ_- の両方、もしくは片方で交換して得られた seq-pair を隣接解とした。しかし、symmetric-feasible でない seq-pair や、separate 制約を満たしていない seq-pair が得られた場合、隣接解の生成をやり直すようにした。このような seq-pair は、生成しない方がより効率良く解を探索する事が出来るため、上記の隣接解生成手法に対称集合に属するセルと属さないセルとの交換操作は行わないなどの工夫を加えた。

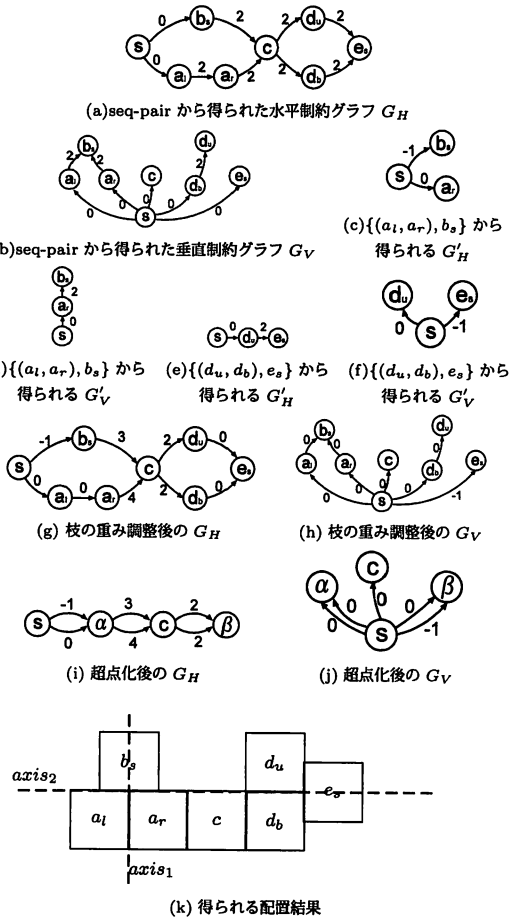


図 6 排他的近接対称配置制約 $\{(a_l, a_r), b_s\}$ と、 $\{(d_u, d_b), e_s\}$ と、symmetric-feasible seq-pair $(b_s, a_l, a_r, c, d_u, d_b, e_s; a_l, a_r, b_s, c, d_u, d_b, e_s)$ が入力されたときの各 step で得られる制約グラフと得られるパッキング (図中の α は $\{(a_l, a_r), b_s\}$ を超点化、 β は $\{(d_u, d_b), e_s\}$ を超点化したもの)

5.2 評価関数

同じ対称集合に属するセルは、出来るだけ小さくまとまって配置された方が、より良い配置と言えるだろう。そのため、それぞれの対称集合の全てのセルを囲む最小の外周矩形の半周長をコスト関数に加えた。

具体的には、 α を固定係数、width, height を配置を囲む外周の幅と高さ、 $length_i$ を i 番目の対称集合の外周矩形を囲む最小の矩形の半周長、 m を対称集合の数とし、

$$Cost = width * height + \alpha * \sum_{i=1}^m length_i \quad (3)$$

をコスト関数とした。

5.3 配置実験

[8] の図 9 で示されている、いずれも垂直軸についての 3 つの対称集合 (対称対 4, 対称対 6, 対称対 2 自己対称 1) を含んだ 65 個のセル集合 (biasynth_2p4g) と、図 10 で示されている、同じく垂直軸についての 5 つの対称集合 (対称対 8, 対称対 3, 対称対 3, 対称対 6, 対称対 2) を含んだ 110 個のセル集合 (lnamixbias_2p4g) を入力として配置実験を行った。

表 1 Balasa らの手法 (Sun Blade 100) [8] と、甲田らの手法 (Pentium4 3.2GHz)、本稿での提案手法 (Pentium4 3.2GHz) の実験結果の比較

| Design | #Cell | #Symmetry groups | 対称配置制約 [8] | | 近接対称配置制約 [3] | | 排他的近接対称配置制約 | |
|-----------------|-------|------------------|------------|----------|--------------|----------|-------------|----------|
| | | | Time [sec] | Area [%] | Time[sec] | Area [%] | Time[sec] | Area [%] |
| biasynth_2p4g | 65 | 8+12+5 | 780.00 | 115.00 | 369.08 | 107.89 | 272.28 | 111.25 |
| Inamixbias_2p4g | 110 | 16+6+6+12+4 | 2823.60 | 109.36 | 521.66 | 108.53 | 441.72 | 119.12 |

biasynth_2p4g では、パッキング率 (配置面積/各セル面積の合計) が 111.25% の配置を 272.28 秒で、Inamixbias_2p4g では、パッキング率が 119.12% の配置を 441.72 秒で得ることができた。得られた配置を図 7, 8 にそれぞれ示す。実験に用いた計算機は Pentium4 3.2GHz である。

また比較のため、対称配置制約だけを考慮した Balasa らの手法 [8] と、近接対称配置制約を考慮した甲田らの手法 [3] も含めて、実験結果を表 1 に示す。[3] は計算機に本実験と同じ Pentium4 3.2GHz を用いて、[8] は Sun Blade 100 を用いており単純には比較できない。

表 1 にあるように、面積が少し増大しているものの提案手法を用いた方がより短時間に制約を満たした配置を得ることができた。

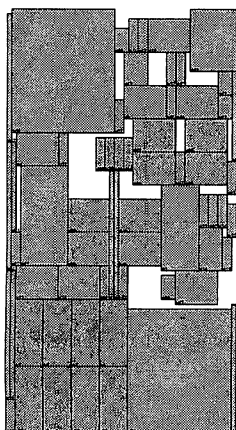


図 7 提案手法での biasynth_2p4g の配置結果 (計算時間 272.28 秒, パッキング率 111.25%)

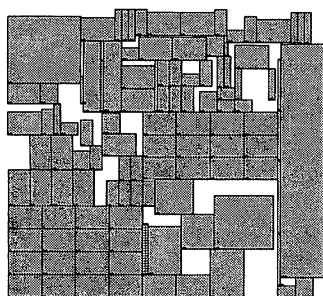


図 8 提案手法での Inamixbias_2p4g の配置結果 (計算時間 441.72 秒, パッキング率 119.12%)

6. まとめ

本稿ではアナログ IC のレイアウト設計において、対称に配置すべきセル対を、出来るだけ対称軸に近く配置させるという近接対称配置制約を、対称集合毎にどのセルも出来るだけ近接するというように制約を厳しくした排他的近接対称配置制約を定義した。そして、排他的近接対称配置制約と、sequence-pair が示唆する制約とを満たした配置の中で、最密なものをセル数 n に対して $O(n^2)$ 時間で得るデコードアルゴリズムを提案した。さらに、提案手法を SA 法探索に組み込み、提案手法が高速に配置を得られる事を確認した。

今後の課題としては、アルゴリズムの改良による高速化、より多くのデータによる実験、などが挙げられる。

文 献

- [1] F.Balasa and K.Lampaert, "Symmetry within the sequence-pair representation in the context of placement for analog design," IEEE Trans.CAD, vol.19, no.7, pp.721-731, 2000.
- [2] J.Cohn, D.Garrod, R.Rutenbar and L.Crley, Analog Device-Level Automation. Norwell, MA: Kluwer, 1994.
- [3] 甲田真一, 藤吉邦洋 "アナログ IC 設計における近接対称配置制約を考慮した配置手法" 信学技報, VLD2006-77, pp.31-36, 2006.
- [4] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani: "Rectangle-packing-based module placement," Proc. IEEE ICCAD, pp.472-479, 1995.
- [5] K.Fujiyoshi, and H. Murata, "Arbitrary Convex and Concave Rectilinear Block Packing Using Sequence-Pair," IEEE Trans. CAD, vol.19, no.2, pp.224-233, 2000.
- [6] S.Kouda, and C.Kodama, K.Fujiyoshi, "Improved Method of Cell Placement with Symmetry Constrains for Analog IC Layout Design," ISPD, pp.192-199, 2006.
- [7] K. Wakata, H. Saito, K. Fujiyoshi, K. Sakanushi, T. Obata, and C. Kodama, "An Improved Method of Convex Rectilinear Block Packing Based on Sequence-Pair," IEICE Trans, Fundamentals, vol.E86-A, no.12, pp.3148-3157, 2003.
- [8] F. Balasa, S.C. Maruvada, K. Krishnamoorthy, "On the exploration of the solution space in analog placement with symmetry constraints," IEEE Trans.CAD, vol.23, no.2, pp.177-191, 2004.