社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2006－SLDM－127 (27)
2006／11／30

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

甲田 真一† 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を提案されてきた．しかし，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いが，これらの手法は対称配置制約のみに対応しており，対称に配置すべきセル対が近くに配置されるとは言えない．そこで本稿では，対称に配置するだけでなく，そのセル対を出来るだけ近くに配置する制約として，近接対称配置制約を定義し，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検証する．

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 Adjacent Symmetry Constraints for Analog IC Layout Design

Shinichi KODA† 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 nearly. Therefore, in this paper, we define "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

近年，携帯電話や無線LANなどの通信デバイスの普及に伴い，アナログICに対する需要が急速に高まっている．アナログICのレイアウト設計では，しばしば指定されたいくつかのセル対を線対称に配置配線する必要がある[1]．その理由は，セルやそれと接続する配線のレイアウトの整合を取ることによって，それによって生ずる寄生素子を含めて整合を取り，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[6]．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．

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，近年では矩形配置表現とSimulated Annealing法とを組み合わせた自動配置手法に対し，対称配置制約を課して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手法を提案した[1]．しかし，Balasaらが提案した手法[1]でデコードすると，セル同士が重なってしまったり，制約は満たすが最密でないパッキングが得られてしまう場合がある．

これに対し，sequence-pairによる制約と対称配置制約は共に線形な式に変換できる事を利用し，線形計画法を用いて最密な配置を求める手法が提案された[3]が，この手法は，問題の規模が大きくなると非常に遅いという欠点があった．

ところで，対称に配置すべきセルの対は，セル特性の整合を取る必要があるが，対称に配置すべきセル対が離れてしまうと，その特性に不整合が生じてしまうことが考えられる．しかし，上記の手法は対称配置制約しか考慮しておらず，対称に配置されたセル同士が離れてしまう場合がある．

そこで本稿では，対称配置制約が課せられるセル集合を対称に配置するだけでなく，それに加えて，できるだけ近くに配置させるという"近接対称配置制約"を定義する．そして，sequence-pairが示唆する制約と，近接対称配置制約を満たした最密な配置を，多角形パッキングの手法を応用する事で高速に求めるデコードアルゴリズムを提案する．また，計算機実験によって提案する手法の有効性を検証する．

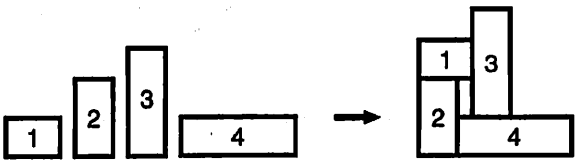


図 1 Packing represented by seq-pair (1234; 2413).

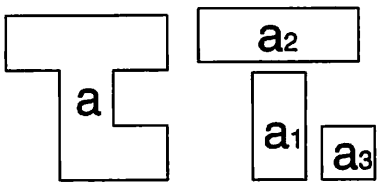


図 2 多角形 a と部分矩形 a_1, a_2, a_3

2. 準　　備

2.1 sequence-pair

sequence-pair [4] (以下 seq-pair) では n 個の矩形の相対位置関係を，矩形名の順列 Γ_+ と Γ_- の対により，$(\Gamma_+;\Gamma_-)$ の形で表す．当然，n 個の矩形の配置について $(n!)^2$ 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 では矩形対の相対位置関係を，以下に示す「上下左右制約」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 (1234; 2413) は，図 1 の配置ような相対位置関係を表す．

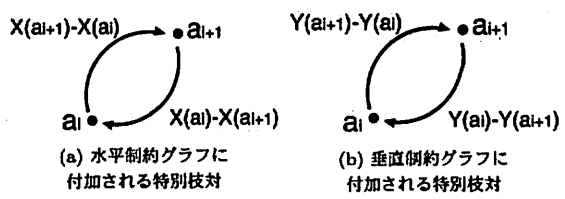
seq-pair が示唆する水平／垂直制約は水平と垂直方向の制約グラフという，点に重みのついた有向無閉路グラフ (DAG) にて表す事が出来る．具体的には，水平制約グラフは各矩形に点が対応してその横幅が点の重みとなり，水平制約関係にある全ての矩形対について，左側の矩形に対応する点から右側の矩形に対応する点に有向枝が張られたものに，source 点と sink 点を付加したものである．垂直制約グラフは，点の重みが矩形の高さで枝が垂直制約関係により張られる以外は同じである．

水平／垂直制約グラフに対して，良く知られた DAG の最長パスアルゴリズムを用いて，source 点から各点への最長パス長を求めて，これを対応する矩形の左下座標とする事により，これらの制約のもとで最も密な左下詰めパッキングを $O(n^2)$ 時間で得る事ができる．

2.2 seq-pair に基づくレクトリニア多角形パッキングを求める手法

外周が水平線分と垂直線分だけの多角形をレクトリニア多角形（以降，多角形）という．[5] では seq-pair を拡張して，凹凸を含んだ多角形パッキングを表現する手法が提案された．本来 seq-pair は矩形のみしか扱えないため，ここでは各多角形は水平／垂直線分で切断され部分矩形の集合として扱われている．多角形を部分矩形集合に分割した例を図 2 に示す．

与えられた seq-pair からその水平／垂直制約に基づいた左下詰めの多角形パッキングを得るための手続き [5] は，まず，seq-pair から矩形パッキングの場合と同様の水平／垂直制約グラフを作る．次に，各点の重みをその点から出る枝に移す．そして，多角形を復元させるために特別枝対と呼ばれる特別な枝を制約グラフに付加する．この枝が付加された節点対に対応する部分矩形対は，枝の重み分の距離で固定される．$X(a_i) - X(a_{i+1})$ を多角形が復元されたときの，部分矩形 a_{i+1} の左辺 x 座標を基準としての，a_i の x 座標である．図 3(a)(b) に水平制約グラフと垂直制約グラフに付加される特別枝対の例を示す．



(a) 水平制約グラフに付加される特別枝対　　(b) 垂直制約グラフに付加される特別枝対

図 3 特別枝対の例

特別枝対が付加された制約グラフ上で source 点から各点までの最長パス値を求め，その値を各部分矩形の左下座標とすることで，seq-pair が示す水平／垂直制約に基づく多角形の左下詰めパッキングを得ることができる．しかし，特別枝対はグラフ上に閉路を作るため，最長パス値の計算に DAG の最長パスアルゴリズムを用いることはできない．そのため，Ford の最短パスアルゴリズムを応用した最長パスアルゴリズムを用いて，最長パス値を計算し部分矩形の座標を決定する．この手法の計算複雑度は，n を部分矩形の数とすると $O(n^3)$ である [5]．

2.3 対称配置制約

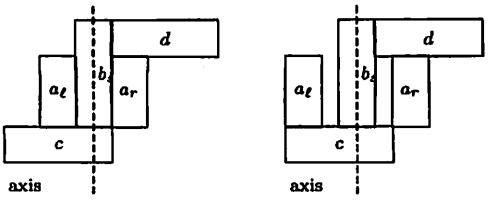
対称配置制約は，複数のセル対からなる対称集合により表され，その中の各セル対を，共通の垂直または水平な軸 (対称軸) に対して線対称になるように配置せよという制約である．対称集合には，それ一つで対称に配置せよという自己対称セルが含まれる場合もある．これはつまり，対称軸の x または y 座標にその中心を合わせなければならないセルである．対称集合は 1 つとは限らず，各々の対称集合毎に 1 本の垂直または水平な軸に対して対称に，その集合のセル対を配置する．

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を左セル，右のセルを右セルと呼び，それぞれ a_ℓ,a_r の様に添字をつけて表す．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を上セル，下のセルを下セルと呼び，それぞれ a_u,a_b の様に添字をつけて表す．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，$\{(a_\ell, a_r), b_s\}$ のように表記する．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. 近接対称配置制約と提案するデコードアルゴリズム

3.1 近接対称配置制約

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るが，そのセル同士を離して配置してしまうと，セル特性にばらつきが生じてしまう．そのため，対称対のセル同士は，できるだけ近付けて配置させることが求められる場合が多い．しかし，対称配置制約だけでは，対称性さえ満たされていればよいため，セル同士が必ずしも近くに配置されるとは言えない．



(a) 近接対称配置制約の場合　(b) 対称配置制約の場合

図 4　seq-pair $(a_\ell\, b_s\, d\, c\, a_r\,;\, c\, a_\ell\, b_s\, a_r\, d)$ と，対称集合 $\{(a_\ell, a_r), b_S\}$ から得られる最密な配置

そこで我々は，"各対称集合における任意の対称対を離して配置してはいけない" という近接対称配置制約を定義する．これは以下の 3 つの制約からなる．

> **近接対称配置制約**
> 近接対称配置制約を与えられたセル集合は，以下の制約を満たす．
> （1） 対称集合毎に，対称対を対称軸に対して対称に配置する．
> （2） 対称集合毎に，対称対を出来るだけ対称軸に近づけて配置する．
> （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モジュールが入り込む事を禁じる．

この制約は配置表現方法が示唆する制約の下で，対称対を対称にかつ，出来るだけ対称軸に近づけて配置させ，対称対の間に，その対称集合に属さないセルが入り込む事を禁じるという制約である．本稿では，対称に配置すべきセル集合はすべて近接対称配置制約が与えられているとし，近接対称配置制約も対称集合で表現する．

近接対称配置制約を課すことで，同じ対称集合に属するセル同士は対称軸を中心に，できるだけ近くに配置することになる．その結果，面積が増大してしまうことがあるが，アナログ IC 設計においては，より小さい面積で配置することよりも，回路性能を上げるために与えられた制約を満たすことの方が重要であるので，本稿ではこのような面積の増大を許容する．

例えば，seq-pair $(a_\ell\, b_s\, d\, c\, a_r\,;\, c\, a_\ell\, b_s\, a_r\, d)$ に，近接対称配置制約として $\{(a_\ell, a_r), b_s\}$ が課せられている場合は，図 4(a) の配置となるが，対称配置制約として $\{(a_\ell, a_r), b_s\}$ が課せられている場合の図 4(b) より図 4(a) は，c と a_r の間に左右制約関係があるために，全体の横幅が大きくなる．

近接対称配置制約 (3) は，seq-pair においては，"Γ_+, Γ_- において，各対称集合に属する対称対の 2 つのセルの間に，それ以外の対称集合のセルが共通に存在している seq-pair を許容しない" という制約とみなすことができる．

3.2 symmetric-feasible seq-pair

本稿では，与えられた近接対称配置制約と symmetric-feasible seq-pair が示唆する制約とを，同時に満たす配置を高速に得る手法を提案する．symmetric-feasible とは，Balasa らが，対称集合が 1 つのとき，対称配置制約を満たす配置が存在する seq-pair の必要十分条件として定義したものであり [1]，それぞれの対称集合について，次式 (1) を満たす事であるとしている．

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

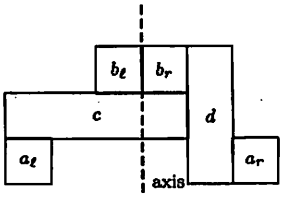


図 5　対称配置制約 $\{(a_\ell, a_r), (b_\ell\, b_r)\}$ を満たしているが，対応する唯一の seq-pair $(b_\ell\, b_r\, c\, a_\ell\, d\, a_r\,;\, a_\ell\, c\, b_\ell\, b_r\, d\, a_r)$ が symmetric-feasible ではない配置

対称軸が垂直線である場合，式 (1) は

$$\Gamma_+^{-1}(x_\ell) < \Gamma_+^{-1}(y_\ell) \Leftrightarrow \Gamma_-^{-1}(y_r) < \Gamma_-^{-1}(x_r)$$

となる．但し，モジュール a と対称に配置すべきモジュールを $sym(a)$ を表す．

この symmetric-feasible は [3] において，必要条件ではなく，図 5 のような配置は表現できないと指摘されている．そして，[1] では SA 法による探索の際に symmetric-feasible sequence-pair だけを探索しているので，この手法は最適解を見落とす可能性があるとしている [3]．

しかし，近接対称配置制約が与えられている場合，[3] で指摘されているような反例は適応されず，symmetric-feasible seq-pair は制約を満たす配置を全て表現することができると考えられる．

3.3 提案するデコードアルゴリズム

提案するアルゴリズムは 2 つのステップ，*1st step* と *2nd step* から成る．1st step では，対称集合毎に属する全てのセルから対称軸までの距離を決定する．2nd step では，1st step で求めた距離を固定させるように，多角形パッキングのアルゴリズムを応用することで，各セルの座標を得る．

まず，1st step では，symmetric-feasible seq-pair と，近接対称配置制約から，各対称集合毎に，対称軸から対称集合に属するセルまでの距離を実現可能な中で最小の値に決定する．図 6 に，対称軸が垂直軸である対称集合 1 つを入力として，相対距離を決定するアルゴリズムを示す．対称軸が水平軸の場合でも同様にして，対称軸から対称集合に属するセルまでの距離を求める事が出来る．

例えば，全てのセルの大きさを 2×2 として，symmetric-feasible seq-pair $(b_s\, a_l\, a_r\, c\, d_u\, d_b\, e_s\,;\, a_l\, a_r\, b_s\, c\, d_b\, d_u\, e_s)$ と，近接対称配置制約 $\{(a_\ell, a_r), b_s\}$，$\{(d_u, d_b), e_s\}$ が与えられた場合，近接対称配置制約 $\{(a_\ell, a_r), b_s\}$ についての G' は図 7(a) となる．このグラフにおいて，*source* から各節点の最長パス長を計算することで，b_s, a_r の対称軸からの相対距離がそれぞれ，$-1, 0$ と決定する．次に，a_r の座標と幅から a_ℓ の対称軸からの相対距離を -2 と決定する．

次に，2nd step では，2.2 で述べた多角形パッキングのアルゴリズムを応用し，Ford のアルゴリズムを用いて，seq-pair と 1st step で得られた距離を保ちながら，各セルの左下座標を求める [5]．具体的には，seq-pair から水平制約グラフ G_H を求め，垂直な対称軸に対応する節点を付加する．次に，対称軸が垂直軸な各々の対称集合について，G_H 上で第 1 ステップで求めた相対距離が固定されるように，対称集合毎に，対称軸に対応する節点と，対称集合の全てのセルに対応する節点間に双方向に特別枝対を張る．また，対称軸が水平軸な対称集合について，対称対同士の x 座標が同じ値となるように，対称対に対応する節点間には，重み 0 の特別枝対を双方向に張る．そして，Ford のアルゴリズムによって source から各節点までの最長パ

```
対称軸が垂直軸の場合の x 方向の相対距離決定アルゴリズム
  Input: symmetric-feasible seq-pair (Γ+; Γ−),
  対称軸が垂直軸の対称集合;
  Output: 対称軸と対称集合の任意のセル
  の相対距離;
  seq-pair から水平制約グラフ G を作り,
  自己対称セル，右セルに対応する節点集合に
  より定まる誘導部分グラフを G' とする;
  G' に source 点を加える.
  foreach(i=G' 上の節点){
    source から i へ,
    i が自己対称セルなら 重み −w(i)/2 の枝を,
    右セルなら 重み 0 の枝を張る;
  }
  foreach(i=(Γ− で左から対称セルを一つずつ)){
    source から i までの最長パス長を求め,
    それを対称軸からの相対距離として確定する;
  }
  foreach(i=(右セルを一つずつ)){
    i と対になっている左セルの対称軸
    からの相対距離 = −(w(i)+(i の相対距離));
  }
```

図 6 対称軸が垂直軸の場合の x 方向の相対距離決定アルゴリズム

ス長を求め，各セルの x 座標とする．y 座標についても同様にして求める．このとき，G_H もしくは G_V に重み和が正の閉路が存在した場合，その seq-pair と与えられた制約を満たす配置が存在しないことがわかる．

先ほどの例の場合，2nd step によって出来る水平制約グラフは図 7(b)，垂直制約グラフは図 7(c) となり，水平制約グラフにおける *source* から各節点への最長パス長が，対応する各セルの x 座標となる．同様に垂直制約グラフ座標についても座標を求めると，図 7(d) の配置を得ることが出来る．

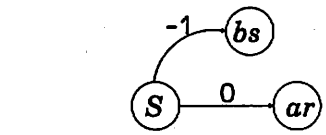
上記の二つの step を経る事で，与えられた近接対称配置制約を満たした配置を，$O(n^3)$ 時間で得る事が出来る．

また，与えられた対称配置制約が一方向の対称軸についての制約だけからなるのなら，対称軸と水平な方向の座標を求めるのには，[3] と同様に，seq-pair から得られる制約グラフを用いて計算する事で高速に座標を求める事ができる．

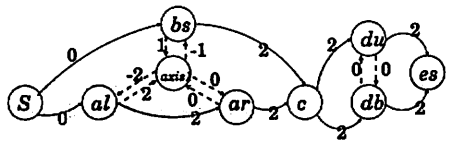
具体的には，対称軸がすべて垂直軸であるなら，seq-pair から y 方向についての制約グラフを求め，対称セルの片方に向かって枝が張られている場合，その対称対の他方にも同じ重みの枝を同じ始点から張る．こうして出来たグラフ上で，source 点から各節点までの最長パス長を求め，対応するセルの y 座標とする．この計算複雑度は $O(n^2)$ である [7].

例えば，seq-pair $(a_\ell\, b\, c\, a_r; b\, a_\ell\, a_r\, c)$ から得られる垂直制約グラフは図 8 の実線で描かれた枝だけによるグラフになる．これに，対称配置制約 $\{(a_\ell, a_r)\}$ が与えられているとすると，枝 (b,a_ℓ) があるので同じ重み $h(b)$ の枝 (b,a_r) を張る．この枝が図 8 で点線で描かれた枝である．

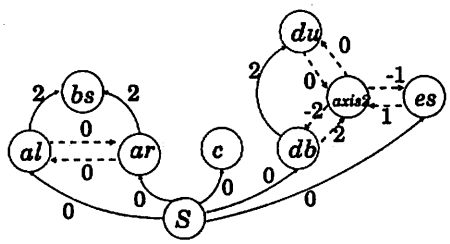
本提案手法に，近接対称配置制約と symmetric-feasible seq-pair を入力した場合，相対距離決定アルゴリズムから得られる座標は，これ以上密に詰める事は出来ないので明らかに最密である．また，多角形パッキングの手法は最密な配置を得る事が出来る．以上のことから以下の定理が導かれる．



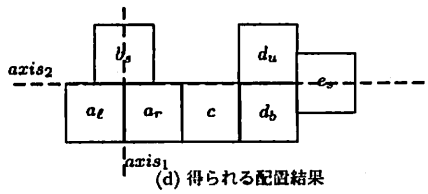
(a) 近接対称配置制約 $\{(a_\ell, a_r), b_s\}$ に対し，1st step で得られる水平制約グラフ G'



(b) 2nd step で得られる水平制約グラフ G



(c) 2nd step で得られる垂直制約グラフ G



(d) 得られる配置結果

図 7 symmetric-feasible seq-pair $(b_s\, a_l\, a_r\, c\, d_u\, d_b\, e_s;$ $a_l\, a_r\, b_s\, c\, d_b\, d_u\, e_s)$ と，近接対称配置制約 $\{(a_\ell, a_r), b_s\}$ と，$\{(d_u, d_b), e_s\}$ が入力された場合のアルゴリズムの例

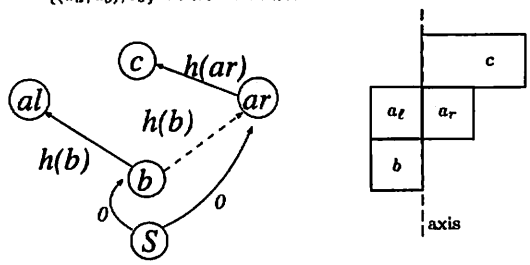


図 8 対称配置制約 $\{(a_\ell, a_r)\}$ と seq-pair $(a_\ell\, b\, c\, a_r; b\, a_\ell\, a_r\, c)$ から得られる制約グラフ．但し $h(a)$ はセル a の高さで S は source 点．

【定理 1】 提案したデコードアルゴリズムに，近接対称配置制約と，symmetric-feasible seq-pair を入力したところ，制約グラフ上に正の閉路が無く配置が得られたのなら，その配置は与えられた制約を満たした配置の中で最密である． ■

3.4 凸多角形アルゴリズムの応用

近接対称配置制約の (3) をさらに厳しくして，

(3') 同じ対称集合に属する任意のセル対の間に，その対称集合に属さないモジュールが入り込む事を禁じる．

という制約とすると，対称軸と垂直な方向だけでなく，平行な方向についても，他の対称集合のセルが入り込むことができなくなり，対称集合はひとつの凸多角形であるとみなすことができる．すると，凸多角形パッキングの手法である [8] の手法を用いることで，$O(n^2)$ 時間でデコードすることが可能となる．

4. 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して SA 法に組み込み計算機に実装した．

4.1 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 を隣接解としている．しかし，symmetric-feasible で無い seq-pair や，対称対間に他の対称集合のセルが入り込んで近接対称配置制約を満たしていない 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．そのため，上記の隣接解生成手法に工夫を加えることで，効率良く解を探索している．

4.2 評価関数

同じ対称集合に属するセルは，でき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る．

具体的には，α を固定係数，$width, height$ を全体の幅と高さ，$length_i$ を i 番め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，

$$Cost = width * height + \alpha * \sum_{i=1}^{m} length_i \qquad (2)$$

をコスト関数として与える．

4.3 配置実験

[2] 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．得られた配置を図 9,10 にそれぞれ示す．実験に用いた計算機は Pentium4 3.2GHz である．

この実験から，65 個のセル集合では，パッキング率 (配置面積/各モジュール面積の合計)107.89%の配置を 369.08 秒で，110 個のセル集合では，パッキング率 108.53%の配置を 521.66 秒で得ることができた．

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 [2] と，線形計画法を用いて対称配置制約を満たした配置を得る手法 [3] も含めて、実験結果を表 1 に示す．[2] は計算機に Sun Blade 100 を，[3] は計算機に Pentium4 3.2GHz を用いている．提案手法の配置実験と，[2] とは実験に用いた計算機が違うため，単純に比較することはできないが，3.1 で示しているように近接対称配置制約が与えられると対称配置制約が与えられている場合に比べ面積が増大してしまう場合があるにも関わらず，表 1 から，提案手法を用いた方が密に詰まった配置を，より短時間に得る事ができた．

5. ま と め

本稿ではアナログ IC のレイアウト設計において，指定されたセル集合を水平，もしくは垂直線に対して対称に配置させるという対称配置制約に加えて，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，近接対称配置制約を定義した．そして，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，最密なものを高速に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る事が出来る事を確認した．

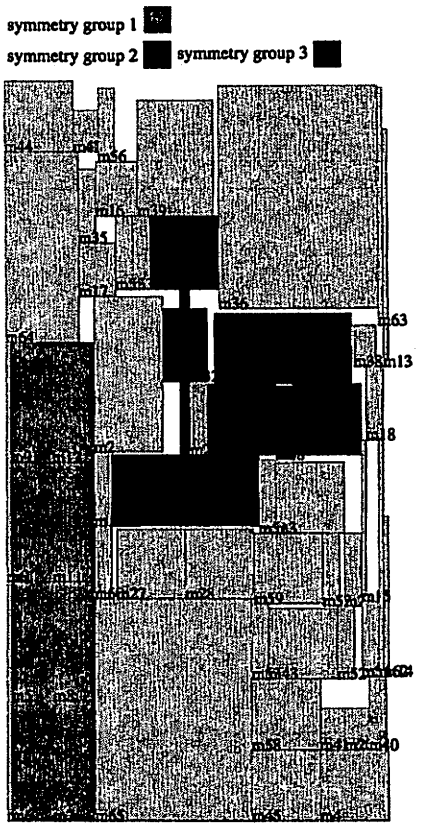


図 9 提案手法を SA 法に組込み，biasynth_2p4g を入力しての配置実験結果 (計算時間 369.08 秒，パッキング率 107.89%)

今後の課題としては，3.4 で示した手法の実装と実験，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．

謝　辞

本研究を進めるにあたって御教授頂きました，東京工業大学岡田健一助手に深謝いたします．

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] 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.

[3] S. Kouda, and C. Kodama, K. Fujiyoshi, "Improved Method of Cell Placement with Symmetry Constraints for Analog IC Layout Design," ISPD, pp.192-199, 2006.

[4] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani:

表 1 提案手法 (Pentium4 3.2GHz) と Balasa らの手法 (Sun Blade 100) [2], ISPD 06 の手法 (Pentium4 3.2GHz) [3] の実験結果の比較

Design	#Cell	#Symmetry groups	Balasa's results [2]		ISPD 06 [3]		Proposed method	
			Time [sec]	Area [%]	Time[sec]	Area [%]	Time[sec]	Area [%]
biasynth_2p4g	65	8+12+5	780.00	115.00	402.60	106.53	369.08	107.89
lnamixbias_2p4g	110	16+6+6+12+4	2823.60	109.36	3252.00	108.58	521.66	108.53

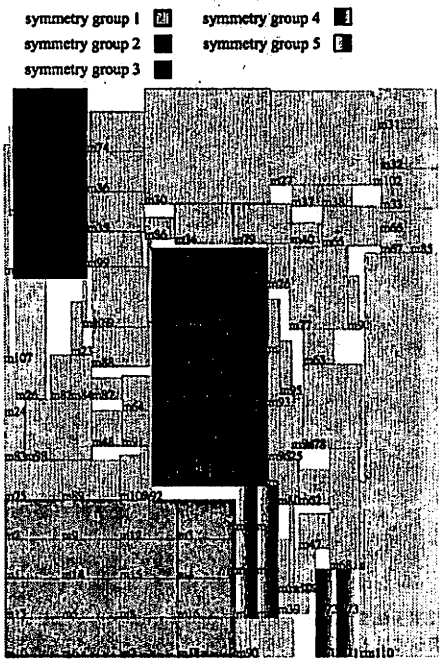


図 10 提案手法を SA 法に組込み, lnamixbias_2p4g を入力しての配置実験結果 (計算時間 521.66 秒, パッキング率 108.53%)

"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] J. Cohn, D. Garrod, R. Rutenbar and L. Crley, Analog Device-Level Automation. Norwell, MA: Kluwer, 1994.

[7] 斎藤宏明, 藤吉邦洋, "L 型ブロックパッキングの高速化に関する研究," 第 13 回 回路とシステム軽井沢ワークショップ, pp.245-250, 2000.

[8] K. Wakata, H. Saito, K. Fujiyoshi, K. Sakanushi, "An Improved Method of Convex Rectilinear Block Packing Based on Sequence-Pair," IEICE Trans. Fundamentals, vol.E86-A, no.12, pp.3148–3157, 2003.