

# 初期配置アルゴリズムが最終配置結果に与える影響

村井真一 柿沼守男 今井正紀 (三菱電機株式会社)

## 1. はじめに

論理装置の実装設計におけるモジュール配置問題、即ちプリント基板上の部品配置設計やLSIチップ上のセル配置設計等の自動化は、通常、初期配置処理に続く逐次改善処理によって行なわれることが多い。

初期配置、配置改善処理に関してM. Hanan 他の実験結果<sup>[1]</sup>は

- ① 初期配置結果が配置改善後の最終結果に影響することを示しており、更に
- ② 組立式初期配置を実行した後、配置改善を行なう方がランダム初期配置結果に同様の改善を行なう場合に比べ通常良い結果が得られることを示している。即ち現状で許容できる処理時間の範囲内では“より良い”初期配置は“より良い”最終配置を与えることが多いと考えられる。

さて一般に行なわれている組立式初期配置<sup>[2]</sup>は、ローカルに最適化することの繰り返しであり、グローバルには歪を伴うと考えられる。故に我々はグローバルに良い結果が期待できる初期配置手法を採用すれば、最終的に良い結果を得られるのではないかと考えた。

ところでE.G. Ulrich 他はクラスタリングとリニア配置により、グローバルなリニア初期配置手法を実現し、更にボトムアップよりトップダウンが良いことを示した。<sup>[3]</sup> そこで我々は文献<sup>[4]</sup>と同様これを2次元に拡張し、クラスタリングによるトップダウン2次元分割初期配置プログラム<sup>[6],[7]</sup>を作成した。

本報告では初期配置アルゴリズムとして組立式とトップダウン式を用いた場合として、初期配置結果及び配置改善後の最終配置結果がどのように異なるかについて報告する。更にブロッキングの効果についても報告する。

## 2. 配置アルゴリズム

### 2.1 クラスタリングによる組立式初期配置<sup>[2]</sup>

このアルゴリズムは、まず他のモジュールと最も強く接続しているあるモジュールを選び、配置エリアの中央に配置する。その後、未配置モジュールのうち既配置モジュール群に最も強く接続しているモジュールを選び出し、それを現状で最良の位置に配置することを繰り返す手法である。接続の強さとして第1に予想配線本数を、第2に外部接続度を考慮する。モジュール $i$ の予想配線本数 $m_i$ を(1)式で、外部接続度 $T_i$ を(2)式で示す。

$$m_i = \sum_{k \in A \cap B} \frac{n_k}{n_{k-1}} \quad (1)$$

ここに  $k$ : 信号(ネット)

$n_k$ : 信号 $k$ に接続するモジュール数

$A$ : モジュール $i$ と既配置モジュールを接続する信号の集合

$B$ : モジュール $i$ と未配置モジュールを接続する信号の集合

$$T_i = \sum_{j \in M_i} CN_{ij} \quad (2)$$

ここに  $M_i$  : モジュール  $i$  と接続するモジュールの集合  
 $CN_{ij}$  : モジュール  $i, j$  間の接続度。(3)式に示す。

$$CN_{ij} = \sum_{k \in I \cap J} p_k \cdot g(k_{ij}) \quad (3)$$

ここに  $I$  : モジュール  $i$  に接続する信号の集合  
 $J$  : モジュール  $j$  に接続する信号の集合  
 $p_k$  : 信号(ネット)  $k$  の重み  
 $g(k_{ij})$  : ネット  $k$  を構成するモジュール  $i, j$  間のエッジ  $k_{ij}$  に付随する重み

モジュール  $i$  を配置すべき最良の位置は、モジュール  $i$  が接続する既配置モジュールの重み付き重心に最も近い空スロットとする。

## 2.2 トップダウン2次元分割配置

### (1). 非ブロッキングモード

クラスタリング トップダウン2次元分割配置を行なう準備として、モジュールをボトムアップに結合していく。即ちモジュール間の結合度を尺度に、ペア単位で結合(クラスタ)することを繰り返す。2つのモジュール  $i, j$  間の結合度  $CV_{ij}$  は(4)式とし、この値が最大であるペアを順次結合する。このようにして結合(生成)されたクラスタは新たな次の結合の対象になりうる。結合過程およびその結果は binary tree (Clustering tree と呼ぶ) として表現できる(図1)。

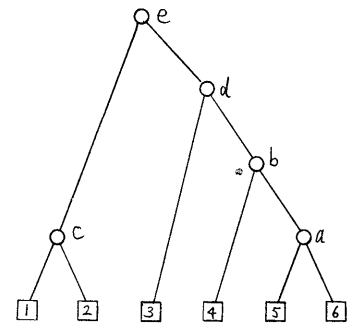


図1. Clustering tree

$$CV_{ij} = f(s_i) \frac{CN_{ij}}{T_i - CN_{ij}} + f(s_j) \frac{CN_{ij}}{T_j - CN_{ij}} \quad (4)$$

ここに  $CN_{ij}$  : クラスタ  $i, j$  間の接続度  
 $T_i$  : クラスタ  $i$  の外部接続度  
 $T_j$  : クラスタ  $j$  の外部接続度  
 $f(s_i), f(s_j)$  : 各々クラスタ  $i, j$  のサイズ(クラスタに属するモジュール数)  $s_i, s_j$  のある関数

トップダウン2次元分割配置 はじめに最上位クラスタを配置エリアの全面に対応させる。以後トップダウンで Clustering tree に従い、エリアのある領域に対応付けられているクラスタを取り出し、その1レベル下位のクラスタサイズに比例させてその領域を2分割し、各々の中心に2つのクラスタを対応付けることを繰り返す(図2)。領域2分割には4通りの可能性があり(図3) そのうち最適な分割を採用する。すべてのクラスタが処理された時に初期配置は終了する。

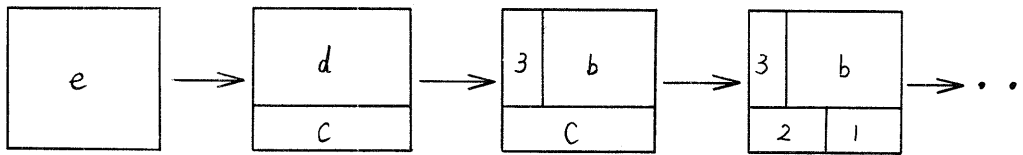


図2. トップダウン2次元分割配置

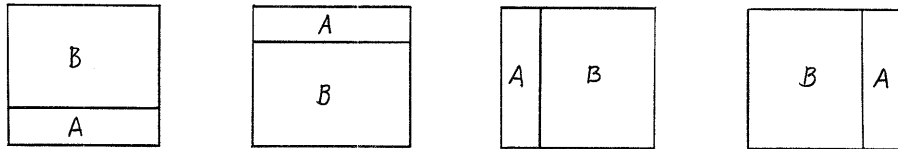


図3. 4通りの2分割

ペア緩和法による配置改善 以上の初期配置手法における i) クラスタリング・アルゴリズムの限界 (pairwise 即ち sequential) および ii) トップダウン配置法の限界 を克服するために、ペア緩和法 (FDR) <sup>[1]</sup> を利用して、配置の改善を行なう。ペア緩和法はペア交換法の1種で、ペア交換法が交換ペアを無作為に抽出するのに対して、ペア緩和法では交換ペアを限定するための heuristics として緩和法の考えを取り入れている。即ちすべてのモジュールについて、それに接続しているモジュールから張力を働かせてその平衡点 (target point) を求め、お互いの現在位置がそれぞれ相手の target point の  $\epsilon$  近傍にあるようなペアについてのみその位置の交換を試みる (図4)。配置が改善される場合のみ実際に交換する。

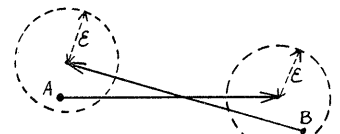


図4. ペア緩和法の概念

(2) ブロッキングモード

配置改善時に複数個のモジュールをブロックとしてまとめて交換する方法があり、それは改善に効果的であることが知られている。<sup>[5]</sup> しかし個々のブロックのサイズが異なる、と、交換処理が難しい。そこであらかじめ配置エリアを大きさの等しい  $N$  個のベンチに区切っておく。一方クラスタサイズがベンチサイズを越えないようにクラスタリング (ブロッキング) して  $M$  個のブロック ( $M \leq N$ ) を生成する。さらに  $M$  個のブロックを最後までクラスタリングする (図5)。

配置は次の4ステップ処理で実行される。

- i) ブロックの初期配置
- ii) ブロックのFDRによる配置改善
- iii) 各ベンチ内でのモジュール初期配置
- iv) 配置エリア内でのモジュール配置改善

- クラスタ
- ブロック
- 内部モジュール
- 外部モジュール

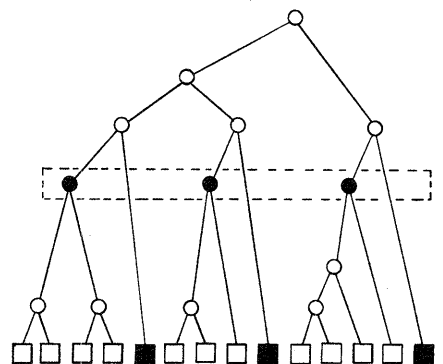


図5. ブロッキング

### 3. 配置エリアの構造, 評価関数

モジュールは内部モジュールと外部モジュールに分けられる。内部モジュールは内部スロットに、外部モジュールは外部スロットに配置することができる。図6に内部スロット、外部スロットの並びを示す。1辺毎の外部スロット数は等しくなくてもよく等間隔である必要もない。内部スロットの行間あるいは列間も同様に不平等でもよい。但し同一行に属する内部スロットのy座標はすべて等しく、また同一列に属する内部スロットのx座標もすべて等しくなければならぬ。

配置結果の“良, 不良”は仮想の総配線長で評価する。線長 $l$ として(5)式を使用した。

$$l = \frac{\sum_{i,j} C_{ij} d_{ij}}{\sum_{i,j} C_{ij}} \quad (5)$$

ここに  $i, j$  : 信号ネットを構成するモジュール  
 $C_{ij}$  : モジュール $i, j$ 間の接続度  
 $d_{ij}$  : モジュール $i, j$ 間の距離

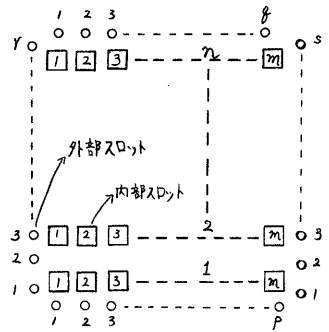


図6. 配置エリア構造

### 4. 実験結果

#### 4.1 初期配置結果の比較

サンプル毎のトップダウン式初期配置結果を100とした場合の組立式初期配置結果を表1に示す。

サンプル	A	B	C	D	E	F
組立式初期配置結果	87	95	101	95	112	97
内部スロット使用率	56	82	75	81	65	66
外部スロット使用率	69	87	93	91	99	91

表1. トップダウン式を規準とした組立式初期配置の結果

表1によると内部スロット使用率は無関係で、むしろ外部スロット使用率に相関があるように見える。すなわち外部スロット使用率が大きくなるとトップダウン式が優位になり、小さくなると組立式が優位になる。外部スロット使用率が大きくなるにつれて組立式の結果が悪くなるのは主に次の2つの理由がと思われる。  
 1) 組立式はモジュールに対しsequential処理で結果は処理順序に依存する。  
 2) 外部スロットは内部スロットの周辺にある。

組立式ではあるモジュールの配置位置を決定する時、それに接続する未配置モジュールには何も考慮を払わない。そこで一般に処理が進むにつれて歪が増加する。さらにここで使用したアルゴリズムではどの外部モジュールも、それが接

繞するすべての内部モジュールよりも必ず遅れて配置が決定される。しかも外部スロットは配置エリアの中央から離れている。それゆえ内部モジュール配置の歪は必ずしも存在せずむしろローカルには良好なに対し、外部モジュール配置の歪は著しいことが考えられる。

一方トップダウン式はその2次元分割処理により内部モジュールの配置密度を均一化しようとするから内部スロット使用率が小さいと組立式に劣るとも考えられるが表1では内部スロット使用率による差は判然としない。試みに特定内部スロットを配置禁止にして仮に内部スロット使用率を100%にしてみても(表2)その結果はほとんど変化しない。そこで内部スロット使用率が60%以上になればやはりトップダウン式の密度均一化は、配置結果に影響しないと考えられる。

サンプル	A	B	C	D	E	F
内部スロット使用率	56	82	75	81	65	66
四隅の配置禁止結果	92	97	106	100	98	101
四辺の配置禁止結果	93	101	103	97	106	102

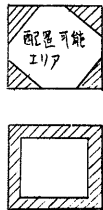


表2. トップダウン式初期配置における配置禁止場所の影響

#### 4.2 最終配置結果の比較

ある一定時間の配置改善効果を表3にしむす。但し組立式初期配置結果を100としてある。

サンプル	組立式の場合			トップダウン式の場合		
	初期	最終	改善率	初期	最終	改善率
A	100	94	6	115	90	22
B	100	89	11	105	86	18
C	100	92	8	99	82	17
D	100	91	9	105	85	19
E	100	82	18	89	70	21
F	100	87	13	103	85	17

表3. 改善率, 最終結果の比較

ここで初期配置結果の良し悪しにかかわらず、トップダウン式初期配置結果は組立式のそれより改善率はもとより最終結果も良好であることが確認された。それは以下の理由によると思われる。

組立式はローカルに調和する配置結果を指向するのに対し、トップダウン式はグローバルに調和する配置結果を指向している。組立式のそのローカルな調和が同じくローカルに改善を試みるFDPRの足がせになり、それゆえ組立式初期配置結果の改善率が高まらなると考えられる。

#### 4.3 非ブロッキング・ブロッキングモードの比較

サンプルFに対してベンチサイズを変えて実行したところベンチサイズが最終結果に影響することがわかった(表4)。

ベンチサイズ	1	10	15	30	60
最終結果	100	95	108	98	140

表4. サンプルFにおけるベンチサイズの効果

ここで得られた最適ベンチサイズを用い、他の5サンプルに対しブロッキングモードの実行を試みた結果、良い場合も悪い場合もあることがわかった(表5)。ブロッキングはブロックレベルのFDPRを可能にする反面、サンプルの自然なクラスタリングを歪ませる。したがってサンプル毎にクラスタリングの歪が小さくなるようなベンチサイズを設定することが、ブロッキングモードの要点であろう。

サンプル	A	B	C	D	E	F
最終結果	89	102	105	99	110	95

表5. 非ブロッキングモードの最終結果を100とした時のベンチサイズ=10の最終結果

#### 5. おわりに

以上の実験から組立式よりトップダウン式がより良い最終配置結果を導くことがわかった。サンプルの平均モジュール数は740個であり、使用計算機はMELCOM CQSMQ 700である。平均CPU timeは組立式では1.3分、トップダウン式では18.1分である。また配置改善は60分で打切った。

トップダウン2次元分割配置プログラムとペア緩和法による配置改善プログラムの組合せは現在マスタースライスLSI設計に配置サブシステムとして実使用中である。

#### 参考文献

- (1) M. Hanan, P.K. Wolf and B.J. Agule, "A study of Placement Techniques", *Journal of Design Automation & Fault-Tolerant Computing*, Vol.1, NO.1, pp. 28-61, Oct. 1976
- (2) M. Hanan and J.M. Kurtzberg, "Placement Techniques", Chap.5 in *Design Automation of Digital Systems: Theory and Techniques*, Vol.1 (Ed. M.A. Breuer), Prentice Hall, N.J., pp. 213-282, 1972
- (3) D.M. Schuler and E.G. Ulrich, "Clustering and Linear Placement", *Proc. 9th Design Automation Workshop* pp. 50-56, June 1972.

- (4) 沖電気, 「パッケージ設計におけるIC配置の一手法」DA夏期シンポジウム資料, 1974年8月
- (5) B.W. Kernighan and S. Lin, "An Efficient Heuristic Procedure for partitioning Graphs", Bell System Tech. J. pp. 291-307, Feb. 1970
- (6) Shinichi Muraï 他 "A Hierarchical Placement Procedure with a Simple Blocking Scheme" 16th. Design Automation Conference pp 18-23 June 1979.
- (7) 村井真一他 「マスタスライスLSIセル配置設計自動化システム」昭和53年度電子通信学会総合全国大会