

シヨート・ノート

論理回路の配置アルゴリズム*

上 田 和 宏**

Abstract

This paper presents a new placement algorithm for allocation of ICs on a printed circuit board or silicon chips on a ceramic board on the criterion of minimizing the total wire length.

The algorithm is based on the concepts of classifying all the elements into two groups of 'friendly elements' or 'unfriendly elements' to one another, supposing attractive forces by friendly elements and repulsive forces by unfriendly elements, and then allocating each element on the position balanced by these two opposite direction forces. The effectiveness of this algorithm has been ascertained comparing with the clustering and the relaxation (or gravitational) method.

1. ま え が き

従来、プリント基板上の IC や、バックボード上のプリント基板の配置の自動化の配置手法として、総配線長最小化を評価基準とした緩和法^{1),2)} (あるいは重心法)が提案され、実用されている。この緩和法では、部品間の接続の強さを、部品間の引力に対応させることによって、極めて短時間で上記の評価基準で良質な配置を得ることを特長としている。しかし、この手法では外部端子の位置結果に大きく影響し、端子が一辺に集中しているような配置構造に対しては適用が困難であるという大きな欠点をもっている。

ここでは、各部品に対して他の全ての部品を '親密な部品' と '疎遠な部品' とに分類し、各部品に対してそれと親密な関係にある部品からは吸引する力を、疎遠な関係にある部品からは反発する力を作用させて、その二種類の力の平衡する位置に各部品を配置するという概念に基づいた配置手法を提案する。本手法では前記の緩和法の欠点を除くことができ、さらに良質な配置を期待できる。

2. 配置アルゴリズム

配置の対象となる部品の集合をグラフにおきかえ、

* A Placement Algorithm for logic modules by Kazuhiro UEDA (Integrated Circuit Section, Musashino ECL, N. T. T.)

** 日本電信電話公社武蔵野電気通信研究所集積回路研究部

n 個のノードからなっているとする。その結合行列をと C とすると、

$$\begin{cases} C_{ij} = m & \text{ノード } i, j \text{ 間に枝が } m \text{ 本あるとき} \\ C_{ij} = 0 & \text{ノード } i, j \text{ 間に枝がないとき} \end{cases}$$

(定義 1) 次のような行列要素 E_{ij} からなる行列を拡張結合行列 E と定義する。

$$\begin{cases} E_{ij} = G(i, j) & \text{for } i \neq j \\ E_{ij} = 0 & \text{for } i = j \end{cases}$$

但し、 $G(i, j)$ はノード i, j 間の最短パスの距離である。最短パスが (i, x, y, j) であるとする、

$$G(i, j) = \frac{1}{C_{ix}} + \frac{1}{C_{xy}} + \frac{1}{C_{yj}}$$

で求められる。例えば、Fig. 1(f) (次頁参照) では $G(1, 2) = 1$, $G(1, 16) = 6$ である。従って、拡張結合行列はグラフの一種の距離行列を与える。

(定義 2) 次のような行列要素 R_{ij} からなる行列を関連行列 R と定義する。

$$\begin{cases} R_{ij} = E_{ij} - E_o & \text{for } i \neq j \\ R_{ij} = 0 & \text{for } i = j \end{cases}$$

但し、

$$E_o = \sum_{i,j} E_{ij} / n(n-1)$$

(定義 3) 集合 M_i, Z_i, P_i を次のように定義する。

$$M_i = \{j | R_{ij} < 0\}$$

$$Z_i = \{j | R_{ij} = 0\}$$

$$P_i = \{j | R_{ij} > 0\}$$

ここで、 M_i, Z_i, P_i の意味を考えてみると、 M_i はノード i といわば '親密な関係にある' ノードの集合、 P_i は '疎遠な関係にある' ノードの集合、 Z_i はそのいずれでもないノードの集合を表わしている。

(平衡処理) ノード i は M_i に属するノードからは $|R_{ij}|$ に比例した吸引力を、 P_i に属するノードからは R_{ij} に比例した反発力を受けるものとする。この2種類の力の相互作用によって平衡する位置にノード i の新しい座標位置を求める処理を平衡処理と呼ぶ。

平衡処理を実現するには種々の方法が考えられるが、ここでは次のような方法をとる。今、着目しているノードを i とする。まず、吸引関係にあるノードの集合 M_i の重心点にノード i を移動させる。この座標を (X_{i1}, Y_{i1}) とすると、

$$\begin{cases} X_{i1} = \frac{\sum_{j \in M_i} |R_{ij}| \cdot X_j}{\sum_{j \in M_i} |R_{ij}|} \\ Y_{i1} = \frac{\sum_{j \in M_i} |R_{ij}| \cdot Y_j}{\sum_{j \in M_i} |R_{ij}|} \end{cases}$$

となる。次に反発関係にあるノードによるノード i の (X_{i1}, Y_{i1}) からの移動分を算出する。この移動距離 $(\Delta X_i, \Delta Y_i)$ をとると、

$$\begin{cases} \Delta X_i = \frac{\sum_{k \in P_i} \alpha \cdot R_{ik} \cdot (K - L_{ik}) \cdot X_{i1} - X_k}{\sum_{j \in M_i} |R_{ij}| \cdot L_{ik}} \\ \Delta Y_i = \frac{\sum_{k \in P_i} \beta \cdot R_{ik} \cdot (K - L_{ik}) \cdot Y_{i1} - Y_k}{\sum_{j \in M_i} |R_{ij}| \cdot L_{ik}} \end{cases}$$

ただし

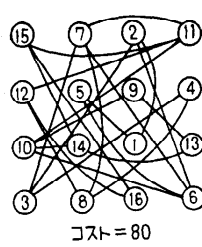
$$\begin{cases} \alpha, \beta = \text{反発係数} \\ K = \text{定数} \\ L_{ik} = \sqrt{(X_{i1} - X_k)^2 + (Y_{i1} - Y_k)^2} \end{cases}$$

と表わせる。ノード i の最終的な座標位置を (X_{in}, Y_{in}) で表わすと、

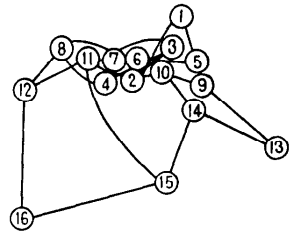
$$\begin{cases} X_{in} = X_{i1} + \Delta X_i \\ Y_{in} = Y_{i1} + \Delta Y_i \end{cases}$$

となる。

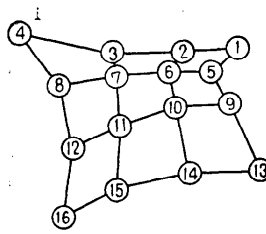
(PG 変換) 平衡処理を全ノードに対して一回ずつ施すのを1サイクルと考え、これを数サイクル繰り返してパターンが収束すると配置処理を終えるが、このままでは各ノードの位置は Fig. 1(e) に示すように格



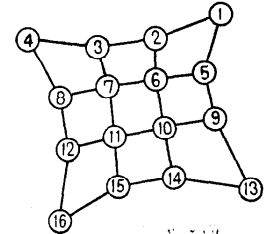
(a) 初期配置



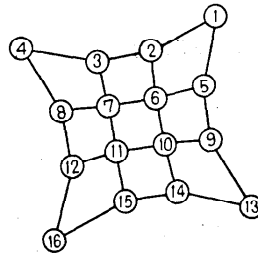
(b) 第1サイクル



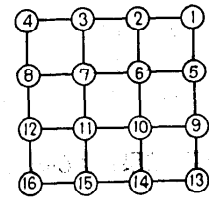
(c) 第2サイクル



(d) 第3サイクル



(e) 第10サイクル



コスト=24

(f) 最終配置

Fig. 1 Example of the placement process

子座標上にはのっていないので、この相対位置関係をできるだけ保存したまま Fig. 1(f) に示すようにノードが占有することが許される格子座標位置に変換する必要がある。この変換をPG変換と呼ぶ。この変換は一つの方法として X, Y 座標で各ノードをソーティングすることによって可能である。

(手法の手順) 本手法の手順を示すと以下のようになる。

- (S. 1) 結合行列 C から拡張結合行列 E を作成し、さらに関連行列 R を作成する。
- (S. 2) 初期配置をセットし、 $K=1$, REP=繰り返しサイクル数とする。
- (S. 3) $i=1$ とする。
- (S. 4) ノード i を選び、平衡処理を行う。
- (S. 5) $i=n$ なら (S. 6) へ、 $i \neq n$ なら $i=i+1$ と

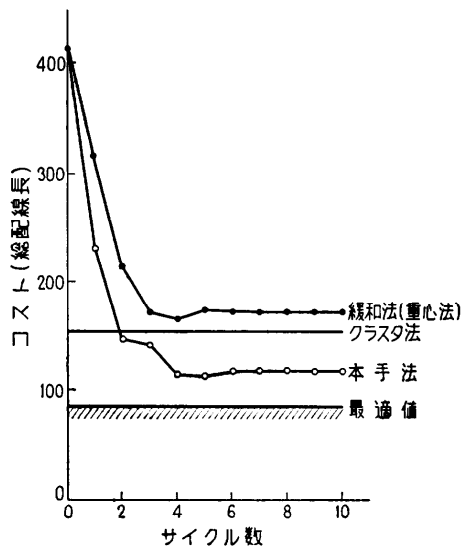


Fig. 2 Change of cost with iteration

して (S. 4) へ。

(S. 6) $K=REP$ なら (S. 7) へ, $K \neq REP$ なら $K=K+1$ として (S. 3) へ。

(S. 7) PG 変換して最終配置を求める。

3. 実験結果および考察

以上に述べた手法にもとづいて実験プログラムを作成し, 手法の有効性の確認を行った. 本プログラムでは端子に相当するノードはないものとし, 平衡処理においては全ノードを同等に扱った. 配置の対象となるグラフとしては, 総配線長最小の意味での最適配置が明らかなものとして Fig. 1 に示すようなグラフによる結果を示した.

Fig. 1 (a)~(f) に処理の途中経過例を, Fig. 2 に処理の繰り返し回数に対するコスト (総配線長) の変化を示す. 配置処理の性質からも予想されるが, Fig. 2 のカーブから収束が極めて速く, しかも最適配置にかなり近い配置結果が得られることが判る.

Table 1 に他手法との配置結果の比較を示す. これから, クラスタ法, 重心法に比べてより最適配置に近い結果が得られていることが判る. 計算機時間は 100 ノードのデータ例で 1 分 50 秒であった. 計算機は DIPS

Table 1 Comparison with other methods

データ番号	部品数	結線数	ランダム配置からの総配線長の改善率 (%)		
			本手法	クラスタ法	緩和法
1	9	8	35	35	24
2	9	12	50	29	50
3	9	20	30	5	23
4	16	20	52	36	37
5	36	60	72	44	41
6	49	84	79	64	60
7	81	144	85	65	63
8	100	180	85	65	74

1 を使用した。

4. むすび

以上の実験結果および考察から, 本アルゴリズムの特徴として次のような項目が挙げられる。

- (1) 部品間の接続情報を十分に配置処理に活かしているため, 最適配置かそれに近い配置が極めて短時間に得られる。
- (2) 初期配置の良否に結果がほとんど依存しない。
- (3) 緩和法とちがって固定端子がないか, あるいは偏ってあっても適用できる。

謝辞 日頃御指導いただく武蔵野電気通信研究所, 向井集積回路室長, 須藤同室長補佐ならびに有益な討論をいただいた集積回路研究室の各位に深謝します. また, プログラムの作成, 実験に当ってお世話になった山本氏に感謝します。

参 考 文 献

- 1) M. A. Breuer: Design automation of digital systems, Chap. 5, Prentice-Hall, New Jersey (1972)
- 2) C. J. Fisk, D. L. Caskey & L. L. West: ACCEL: Automated Circuit Card Etching Layout, Proc. IEEE, Vol. 55, No. 11, pp. 1971~1982 (1967)
- 3) 上田, 山本: マルチチップ LSI 配置の一手法, 昭 49 信学全大, 547 (1974)
- 4) K. Ueda: Placement algorithm for logic modules, Vol. 10, No. 10, pp. 206~208 (1974)
- 5) 上田: 論理回路の配置アルゴリズム, 信学会電子計算機研資, EC 74-34 (1974)

(昭和 49 年 8 月 15 日受付)

(昭和 50 年 7 月 9 日再受付)