

プログラマブルコントローラ用論理図の自動レイアウトプログラム

福田 光子、山田 直之、高野 芳行*

日立製作所 電力・電機開発本部

*日立製作所 大みか工場

プログラマブルコントローラのプログラム設計図である論理図を対象とした自動レイアウトプログラムを開発した。良いレイアウトの指標として、配線の交差、直線性、配線長、分岐方向の4項目を採用し、論理図に適するよう修正を加えた「重心法」、「迷路法」でレイアウトする。即ち、「重心法」は信号の分岐方向を限定するための偏向した重心を求めるように修正し、配線には、障害物の角を格子点とする変則的な「迷路法」を用いた。配線経路は2点間の経路を個別に計算するが、配線順序を工夫することによって、異なる信号間の経路の重複を避ける際に不必要な交差が発生することを回避するとともに、同信号の配線が樹形に分岐するように誘導し、「読みやすい」レイアウトを実現した。

Logic Diagram Automatic Layout Program For Programmable Controllers' Software

Mitsuko Fukuda, Naoyuki Yamada, Yoshiyuki Takano*

Hitachi, Ltd. Power & Industrial Systems R & D Division

*Hitachi, Ltd. Omika Works

The Logic Diagram Automatic Layout Program for the programmable controllers' has been developed. It evaluates the layout in terms of signal directions, crossing of lines, straightness of lines, and line length. Layout procedures are based on "Barycenter Method" for cell layout, and "Maze Method" for routing. These procedures are modified to be effective for the logic diagrams. The ordering procedure for routing is developed to avoid unnecessary crossing and to control branching of lines.

1. はじめに

プログラマブルコントローラ上で実行される制御用のソフトウェアは上位仕様から段階的な詳細化を経て、制御信号を入出力とする制御プログラムとして設計される。これらはラダー図や機能ブロック図、論理回路図等の図式的表現を使って設計されることが多い。

図式的表現は、テキストに比べてプログラム全体が見通しやすいことや、段階的な詳細化がしやすい等の利点があるが、設計図書として利用するには、読みやすくレイアウトされている必要がある。

プログラム設計図として用いられる図式的表現のなかで、論理ゲート等のネットワークで表される論理図は、ラダー図等の表現に比べて信号の流れが分かりやすい反面、ラダー図が一方方向に並べるだけできれいにレイアウトできるのに対して、論理図は簡単にはレイアウトできない。そのような論理図を、パソコン上で簡単に自動レイアウトするプログラムを開発した。

2. 既存のレイアウト手法

レイアウト問題はLSIの分野で古くから研究されており、セル配置を決定する「逐次改善法」^[1]「重心法」^[2]「ペア交換法」^[1,2]、配線経路を決定する「迷路法」^[3]「線分探索法」^[4,5]「グリッドレスルータ」^[6]等、数々の手法がある。

しかし、論理図とハードウェアでは、課せられる制約の性質が異なる。ハードウェアでは、セルの配置間隔、配線の敷設間隔、平行配線長の制限等々の制約を考慮する必要があり、同時にそれらは回路の良否を表す定量的な指標となるが、論理図には成立に必要な不可欠な物理的、本質的制約がない。反対に、人間にとっての「良い」レイアウトという、感覚的な品質が要求される。これまで提案されたシステム^[7,8]では、「良い」レイアウトの指標として、セルの密度、論理の進行する方向、配線の直進性、配線の交差数や、配線の合流・分岐点のレイアウト規則等を用いている。

3. 論理図への適用性

プログラマブルコントローラ用の論理図の例を図1に示す。論理図の要素は、少なくとも1つの入出力端子をもつセルとその間の配線である。フィードバックは同じアドレスを指す入力端子と出力端子で表すので、図面上に配線の戻りはない。ここからプログラムを生成する時に、演算順序のキーとしてセルの位置を利用するので、セルは論理の進行に沿って配置されている必要がある。

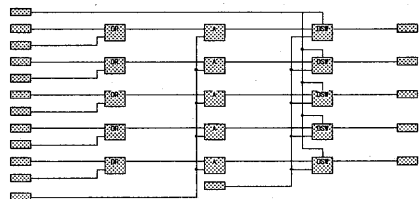


図1 プログラマブルコントローラ用の論理図

論理図におけるセルの形状は、端子数に応じて縦方向に伸長する方形で、横幅はどれもほぼ同じである。端子は原則としてセルの左右にあり、例外的に上辺に1端子をもつ場合がある。通常、セルの各端子はそれぞれに機能的な意味を持っており、端子順序の変更はできない。

このような論理図の「読みやすさ」とは論理の内容が把握しやすいということである。この論理図を人間が「読む」場合、セル間の配線をたどって入出力信号の流れを追う。「読みやすい」ことは「信号の流れを把握しやすい」ということと考えると、良いレイアウトの重要な性質として、配線の交差、直線性、配線長、分岐方向の4点が考えられる。そこで、これらを評価しながらレイアウトするのに適した手法として、セル配置には「重心法」、配線は変則的な「迷路法」を採用した。

以下、論理図の自動レイアウトの方法を説明する。

4. セルの自動配置

まず、セルは論理の進行に逆行しないよう、ランダムな状態から図2のように、接続関係でブロック化する。入力素子から出力素子への経路上のセル数が、どの経路でも同じになるよう、セルの疎らな部分にダミーセルを挿入し、入出力素子のブロックを左右の端に固定すると、同一ラベルごとに縦に並べてセルをブロック化することができる。

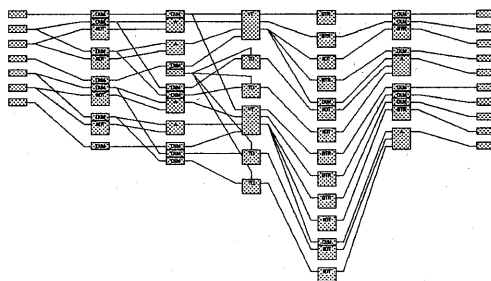


図2 セルのブロック化

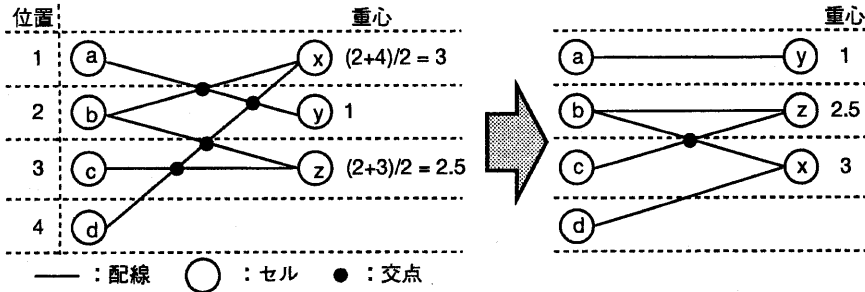
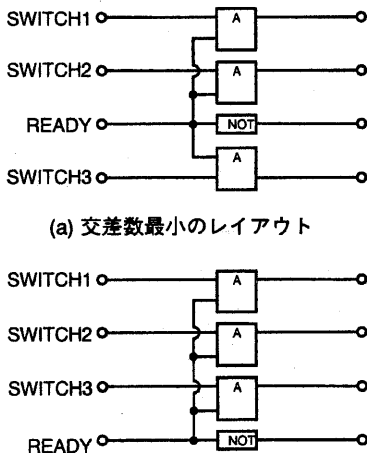


図3 重心法

次に、隣接するブロックのセルを対象に「重心法」で配置する。図3にその例を示す。接続する相手セルの重みの平均を重心とし、重心に従ったセル配置をすることで、セル間の配線の交差を減らすことができる。「重心法」は、他の改善方法に比べて計算効率は良いが、ここで適用にはいくつかの課題がある。

第1に、セルには位置の入れ替え不能な複数の端子があるため、配線に端子配置による局所的な制約がある。端子配置を考えずに重心を計算すると、端子の位置関係から交差が起こりうる。しかし、端子毎に重心を求めても、端子位置が入れ替えられないので重心の通りの配置ができない。



(a) 交差数最小のレイアウト
(b) 交差数の多いレイアウト
図4 交差数の違うレイアウト

第2に、読みやすさと幾何学的な美しさは必ずしも一致しない。例えば図4の場合、(a)は交差が少なく、幾何学的に美しいが、交差の多い(b)の方が、信号READYとその他の信号の関係が分かりやすい。このように、論理図の読みやすさを考えた場合、2方向に分岐して交差を減らすよりも、交差が減らなくても同じ方向に分岐させる方が、流れを追いやすい。

そこで、この2点を解消できるような重心が求まるよう、アルゴリズムを修正した。修正した「重心法」では右ブロックと左ブロック(図5)のセルで、異なる重心の計算方法をとる。

まず、右ブロックのセルの重心は、左ブロックの出力端子毎の重みの平均とする。重みをセル単位ではなく端子単位に付与することで、端子配置の制約を考慮する。

一方、左ブロックのセルの重心は、重みの平均ではなく、分岐方向を決定する1端子の重みで計算する。即ち、接続する相手入力端子の位

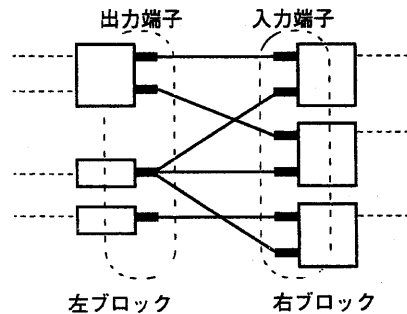


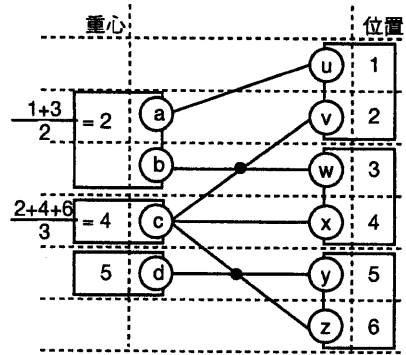
図5 セルブロック対

置が相手セルの中央より上側にあるか、下にあるかという傾向を判断し、上側の端子が多ければ、そのうちで最も上にある端子の重み、反対ならば、最も下の端子の重みを重心とする。接続元になる出力端子が複数ある場合には、接続先の多い端子からセルの重心を計算する。重心を平均値でなく偏った値とすることで、信号の分岐方向を一方向に制限する。例えば、図5で左ブロックのセル配置を決定するとする。「重心法」の場合、端子cの重心は接続先端子v, x, zの重みの平均で4となり、図6(1)のように、配線は上下に分岐する。一方、偏向させた重心を計算した場合、端子cからの接続端子v, x, zがいずれもセルの中央より下にあるので、重心は最も下の端子zの重みから6になる。一方に強い重心を持つので、図6(2)のようになり、分岐方向を一方向に制限できる。

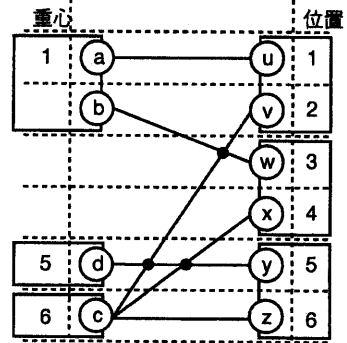
4.2 配線経路の決定

配線は、変則的な「迷路法」で行う。「迷路法」では領域を格子分割するが、ここでは、格子分割せずに各セルの角、およびセルの辺の延長と始点・終点からの垂線の交点だけを格子点とする。論理図の場合、配線の障害物はセルのみで、複雑な形にならないので、2点間の配線経路はこのような荒い分割で十分障害物を回避できる。また、水平方向で始点と終点の間に位置するセルの角だけを格子点にすることで、横方向の配線を2点間の内側に限定する。(格子点がなければ、経路の候補にならない。)ここで、便宜的に各セルの幅を統一すれば、問題は更に簡単になる。セル幅が一定なら、ブロック化によってセルの角は全て一直線上に並び、ブロック間の配線を、障害物の無い、単純な方形内の経路探索にできる。

配線は、終点から始点に向かってマンハッタン距離に基づいたラベルを計算し、始点から終点に向かってラベル最小の格子点をたどる経路



(1) 平均で求めた重心による配置



(2) 偏向した重心による配置

— : 配線 ○ : 端子 ● : 交点

図6 平均した重心と偏向した重心による配置

を求める。マンハッタン距離最小の格子点の列が複数ある場合は、配線の直進性を考慮して選択する。直線的な経路の指標としては、経路の直線区間の最長や二乗和、経路を構成する直線区間の数等が考えられる。ここでは区間数を指標とし、区間数の等しい経路については始点から最も長い水平区間が続く経路を優先する。

複数の配線経路を決定するには、他の配線経路との重複を避ける処理が必要である。重複を避ける方法としては、先に配線された経路を新たな障害物として、配線アルゴリズムで重複を避ける方法や、重複した配線の順序を入れ替えて再配線する方法がある。いずれにしても、格子点が増える、組み合わせ問題になるなど、計

算量の増大が予想される。そこで、経路が重複する場合は、経路長を悪化させない範囲で後から決定した経路を変更し、重複を解決する方法をとる。しかし、この方法では、計算量は増えないものの、どの配線から経路を決定するかという順序がレイアウトに強く影響する。

また、配線の分岐は、できるだけ経路を共有する楕形がよい。単なる2点間の配線アルゴリズムでは、同信号の配線が必ず楕形の分岐をするとは限らない。これを制御するために、ルールとして分岐の形を記述しておく方法¹⁴⁾もあるが、ここでは配線順序を適正化することによって、不必要な交差の解消と適正な配線の分岐を誘導する。

4.3 配線方法から導かれる配線順序

図7は、配線の始点と終点を直結すると、1箇所の交差もおきないセル配置である。この場合、一見して(d)→(c)→(b)の順で配線すれば配線経路は交差しないことが分かる。ここで、各配線の最適経路が通る領域を考えると、図8のようになる。(b)の配線領域は(c)と1辺を共有し、(c)と(d)は重複する部分をもつ。最適経路の領域が2端子を頂点とする方形なら、経路選択の規則から、経路は配線領域の上または下の辺を通ることになる。つまり、始点が終点より下側にある場合は領域の右下、上側にある場合には右上が、最適経路の通過する辺になる。従って、個別に求めた配線経路は各領域の右下(点線(b)~(d)の下側の領域)を通り、領域を共有する(b)と(c)、(c)と(d)の経路が重複することが予想される

まず、(b)と(c)を考えると、(b)の経路は領域の右下辺を通る。次に(c)の経路を求めると、これは領域の上辺で(b)と干渉する。しかも、(c)の終点が(b)の折点になる(図9)ので、(c)の終点から経路を水平/垂直どちらに延長しても(b)と重複し、回避可能な経路がない。つまり、

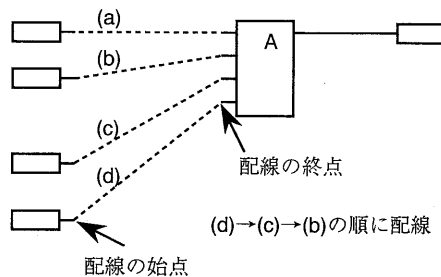


図7 交差のないセル配置の例

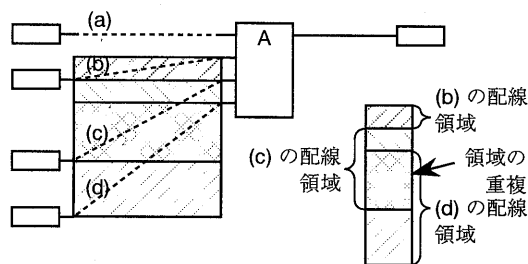


図8 最適な経路の通過する領域

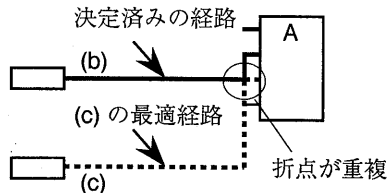


図9 折点の重複

(b)の決定後に(c)は配線できない。

また、(c)と(d)では、(c)を先に配線すると、(d)は(c)に配線領域の重複する右辺を使われている(図10)ので、重複しない経路に修正するには、(d)の経路の垂直成分を領域の右辺以外にずらす必要がある。こうして修正された経路(図11)には、セル配置上は解消された交差がおこる。

配線不能な図9の場合も、配線が交差する図11の場合も、原因は先に配線された(b)、(c)の領域に、あとから配線される(c)、(d)の終点が含まれることに原因がある。従って、先に処理する配線の領域が、他の配線の終点を含まないようにすれば回避できる。しかし、全ての配線

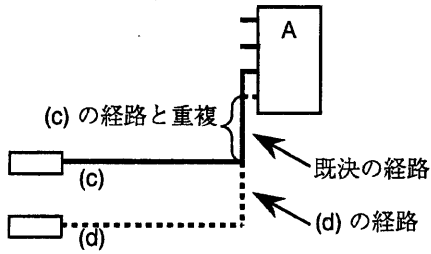


図10 区間の重複

について先に最適経路の領域を求め、領域と各配線の終点の干渉を判定して、優先順序を決定するのは効率が悪い。

そこで、配線経路を計算せずに、配線順序を決定する方法として、縦方向の配線の向きと終点位置による判定を行う。

図7のように配線の始点が終点より下側にある場合は領域の右下、始点が終点の上側にある場合には領域の右上が、最適な経路の通過する領域になる。(セル幅を統一していない場合、配線領域は単純な方形ではなく、頂点を共有する方形の連なりになるが、この処理では便宜上1つの方形で近似する。)配線経路に影響するのは、同じブロック間の、始点から終点の縦方向の配線の向きが同じで、配線領域が重複する場合の配線順序だけである。(以下、始点から終点の、縦方向の向きを配線方向と呼ぶ。)領域が重複しても配線方向が逆であれば、図9のような領域の角の重複がないので配線不能になることはない。

配線方向が同じ場合にだけ、配線の終点が、配線領域の右上または右下の三角形の領域に含まれるか判定し、他の配線の終点を領域に含まない配線の経路を先に決定すれば良い。ここで、配線の終点が同時に配線領域の頂点であることを利用すると、領域が終点を含むかという判定は、終点座標の垂直線分の比較で代用でき、点と領域の関係よりも更に簡単に計算できる。

また、図10のように配線経路上の区間が重な

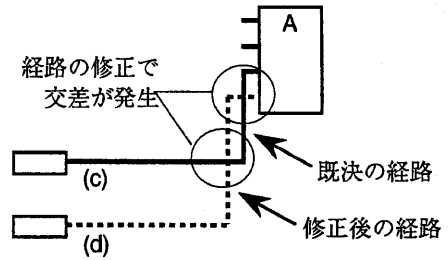


図11 重複の回避から起こる交差

る場合、同信号の配線ならむしろ経路の重複は楕形の配線を導く。そこで、経路重複を避ける配線順序とは逆順で、同信号を連続して配線することにした。これにより、重複した経路を持つ楕形の配線になることが期待できる。

ただし、同信号の配線を連続して処理しても、それぞれの経路は個別に決定するため、折点の少ない適切な経路があればそちらが選択される。そのような経路に対しては、全配線の決定後に経路の統合を試みる。経路の統合は、同信号の経路について、一旦決定した各配線経路を直線区間に分解し、短い区間をより長い区間に併合することで行う。区間の併合は、経路上の区間がより長い区間と交点を持つ場合、その交点を新たな折点として、交点から始点までの経路を、相手区間を含む経路に統合する。(図12) また、交点を持たない場合、直線区間の方向を配線方向に沿って90°回転させた場合に、現在の区間長より短い距離でより長い区間と交点を持てば、元の経路に代えて、方向を変えた新たな区間で経路を変更する。(図13) これにより、

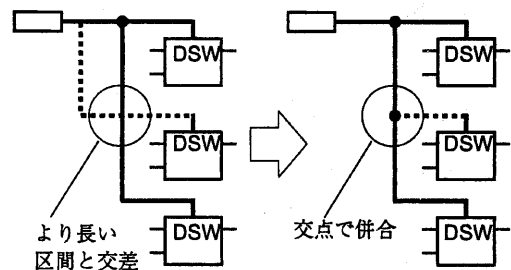


図12 経路の統合 (その1)

統合される配線の経路単体の属性としては折点が増加して直線性が損なわれるが、同信号の配線全体の総延長がより短い楕形の配線でレイアウトされる。

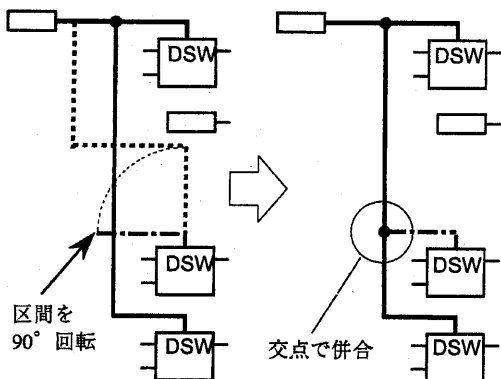


図13 経路の統合 (その2)

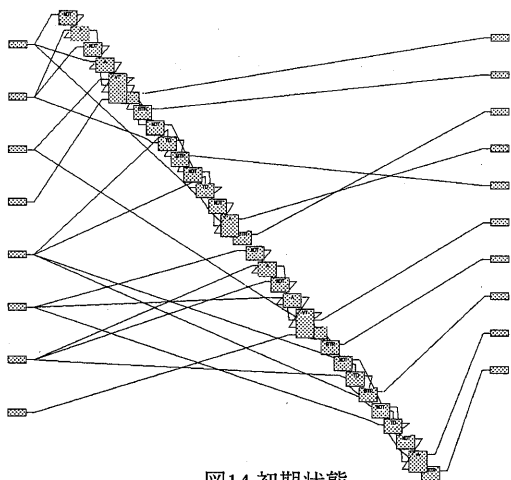


図14 初期状態

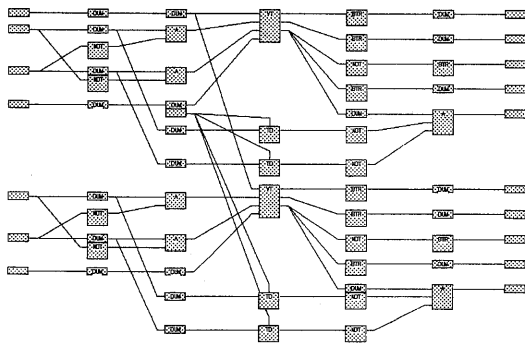


図15 セル配置結果

5. 実行例

レイアウトは、任意の初期状態 (図14) から、まずセルのブロック化を行う。ブロック毎に整列されたセル配置は、偏向した「重心法」で改善する。セル配置の改善は、右端のブロックから、左ブロックのセル配置を更新しながら、左端に向かって1対ずつ実施する。左端のブロックに到達したなら、右ブロックのセル配置を更新しながら、右端のブロックまでセル配置の改善を行う。これを、全ブロック間で配線の交差がなくなるか、あるいは1往復の処理を行うと元の配置に戻るような状態になるまで続ける。

交差がなくなれば、それが最適なセル配置である。そうでない場合、全体の交差数が極小になる配置を示して、ユーザの判断を仰ぐ。現状の配置では不十分だと判断した場合、ユーザは改善すべきブロック対を選んで、個別にセル配置処理を呼び出し、適当な配置が得られるまで繰り返す。

適切なセル配置に到達した後は、端子間を直線で配線できるようにセルの座標を決定する。座標を調整した結果 (図15) に対して自動配線を行い、レイアウトを完了する。自動配線後、ダミーセルを消去した最終的な結果を図16に示す。その結果は、従来の手作業によるレイアウトと比べても遜色の無いものになっている。

この例では、セル配置改善の打ち切り時期をユーザが判断する必要があるが、それ以外は全て自動でレイアウトする。より簡単な論理図では、ユーザの介入無しに初期状態から全自動でレイアウトできる。実際のプログラマブルコントローラ用論理図約100枚に適用した結果では、ユーザの入力を除いた処理時間は、いずれも1秒以内 (CPU: Pentium®133MHz) であった。

Pentiumは、米国Intel社の登録商標です。

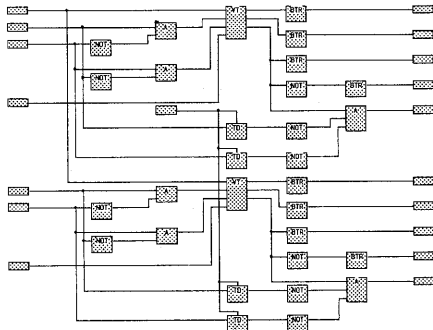


図16 レイアウト結果

6. まとめ

プログラマブルコントローラのプログラムの設計図である論理図を対象とした自動レイアウトプログラムを開発した。良いレイアウトを得るために重要な性質として、配線の交差、直線性、配線長、分岐方向4項目を指標とし、プログラマブルコントローラ用の論理図に適するように修正を加えた「重心法」と「迷路法」で自動レイアウトを行う。

但し、「重心法」は信号の分岐方向の限定するため、信号の流れに沿って偏向した重心を求めるように修正し、配線は、障害物の角を格子点とする変則的な「迷路法」とした。配線経路は2点間の経路を個別に計算するが、配線順序の適正化によって、異なる信号間の経路の重複を避ける際に不必要な交差が発生することを回避し、また、同信号の配線が楕形に分岐するように誘導し、「読みやすい」レイアウトを実現する。

実際の論理図についてこの自動レイアウトを適用した結果、一部でユーザの判断を必要とするものの、人手によるものと同等のレイアウトを、いずれも1シート1秒以内で生成できた。

7. 参考文献

- 1) C. J. Fisk: Proc. IEEE 55, 11, 1971 (1967).
- 2) L. Steinberg: SIAM Rev. 3, 1, 37 (1961).
- 3) C. Y. Lee: An algorithm for path connections and its applications, IRE Trans. Electronic Computers, EC-10, 3, 346 (1961).
- 4) K. Mikami, et. al.: Proc. IFIP 1475 (1968).
- 5) D. W. Hightower: A solution to line-routing problems on the continuous plane, Proc. 5th Design Automation Workshop, 1 (1969).
- 6) 小島 郁太郎、他：多角形領域上の最短経路アルゴリズム, 信学技報, CAS83-205, 45 (1984)
- 7) 新舎、他：ゲート論理図における見易さの基準と実現方法, 情処論, 30, 9, 1219 (1989)
- 8) 下平、他：回路図における素子の自動配置, 第35回情処全国大会, 2437 (1987)