

階層的レイアウト工程全自動化システムの試作

山内貴行 玉柏和男

yama@edag.ptdg.sharp.co.jp

シャープ株式会社 精密技術開発センター

〒632 奈良県天理市櫛本町2613-1

大規模LSIのレイアウト設計では、階層的レイアウト手法が一般的に用いられる。階層的レイアウト手法は、下位階層より順に各階層の実現に適したレイアウト手法を適用する。自動レイアウト技術の成熟により、各階層の設計は自動化が進んでいるが、工程全体では依然人手による最適化作業が必要である。本論文では、階層的レイアウト工程での最適化手法を提案する。詳細な配線領域見積りによる目標形状の決定、目標形状とレイアウト結果との誤差の最小化、レイアウト結果にもとづくブロック位置の調整の各手法を提案する。さらに、手法を当社レイアウトシステムに実装し、階層的レイアウト工程全体を自動化するシステムを試作した。試作による評価では、従来以上のレイアウト品質が得られ、生産性の向上とコスト低減を実現できることが確かめられた。

A Study for Optimizing Hierarchical Layout Design

Takayuki Yamanouchi, Kazuo Tamakashi

Precision Technology Development Center, SHARP Corporation

2612-1, Ichinomoto, Tenri, Nara, 632 Japan

We propose new optimization methods for the hierarchical layout design methodology. The motivation of our approaches is the inconsistency between early floorplanning and detailed one. This inconsistency used to be solved manually. Optimizing the shapes and location of the blocks in lower hierarchy automatically, we can reduce the waste area and improve the productivity. Optimizations are based on the wiring area estimation on the detailed floorplan after fixing the pin positions, and shape functions with wiring area. In order to fit the shape of resulting layout, we propose a new error function. Experimental results show the efficiency of our approaches.

1 はじめに

大規模LSIのレイアウト設計においては、階層的レイアウト手法が一般的に用いられている。LSIのレイアウト全体を大局的に最適化するフロアプランニング設計が完了すると、その結果にもとづいて下位階層より順に、各階層の実現に適した自動レイアウト手法が適用され、全体のレイアウト設計が完了する。各階層のレイアウト設計は、スタンダードセル方式レイアウトやモジュールジェネレータなどの自動レイアウト技術を適用することができる。

また、レイアウトが完了した下位階層のブロック間を接続するマクロセル間配線についても自動配線手法が適用できる。

しかし、階層的レイアウト工程全体としては、依然として人手による最適化作業が残されている。下位階層のレイアウト結果を評価し、その上位階層のレイアウトに進む前に、下位階層のブロック形状を変更したり、ブロック位置を調整し、最適化する作業である。このような作業が必要となるのは、レイアウト結果がフロアプランでの予測と異なるためである。誤差の主な発生要因は以下の理由による。

- ブロック形状は、レイアウト階層構造や、ブロックの概略配置と同時に、チップ全体の大局的な最適化段階で決定される。この時点では端子位置は未決定で、端子位置の影響が考慮できない。また、ブロックの面積自体も端子の分布により変動する。
- フロアプランニング設計で用いる未設計ブロックの面積は予測であり、レイアウトをおこなわず全く誤差なく面積予測を行う方法は分かっていない。

フロアプランは、レイアウトの予測にもとづいて全体の最適化が図られている。また、フロアプラン結果より得られた配線遅延をもとに、タイミング設計をおこなう

場合、できる限りフロアプラン通りにレイアウトする必要がある。したがって、設計者は、下位階層のレイアウト結果にもとづき、元のフロアプランを尊重しながら最適化をおこない、さらに上位階層のレイアウト作業を続ける。

この人手による作業は、レイアウト設計の省力化を妨げるとともに、自動レイアウト手法の知識を必要とするため、システム設計者によるすべてのLSI設計の実現を困難にしている。

本論文では、階層的レイアウトで必要な最適化をおこなう手法を提案する。最適化は3段階から構成される。まず、端子位置決定後の詳細な配線領域見積りにもとづき下位ブロックのレイアウト目標形状を決定する。次に、下位階層をレイアウトし目標形状との誤差を最小化する。最後に、下位階層のレイアウト結果にもとづきブロック位置の調整を図る。

最適化手法を従来の自動レイアウト技術と結合し、階層的レイアウト工程を全自動化するシステムを試作した。試作システムによる評価では、レイアウト品質を従来手法と比較し、最適化手法の有効性を示す。

2 最適化手法の概要

本システムでは、スタンダードセル方式によるブロックと、ROM、RAM、CPUコアなどの既設計のマクロセルからなる階層的レイアウトを想定している。フロアプランでのブロック配置は、スライス構造となるように自動生成する[1]。図1に従来の階層的レイアウトの設計工程と、今回試作したシステムによる設計工程を示す。太線で示された箇所は今回自動化した工程である。

第1章で述べたように、従来の人手による最適化工程では、レイアウト結果を評価し、再レイアウトをおこない、ブロック位置を変更する。

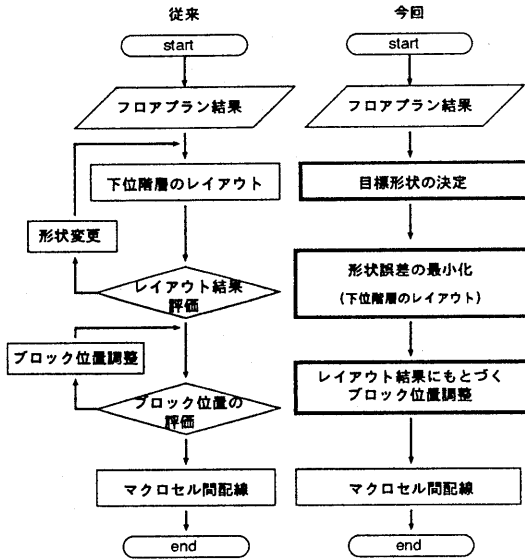


図 1: 階層的レイアウトの設計工程

今回提案する手法では、次の 3 つの処理により最適化をおこなう。

1. 目標形状の決定

初期のフロアプランニングで決定された形状を、詳細な配線領域の予測にもとづいて修正し、レイアウトでの目標の形状とする。ブロック形状は、配線領域の予測値を加えた形状関数の合成手法により決定される。

2. 形状誤差の最小化

下位ブロックをレイアウトする。このとき目標形状とレイアウト結果との誤差を最小化する。新しく、**形状の余裕**と、**目標形状と結果の誤差**を定義する。誤差が減少するように形状を変更してレイアウト結果を改良し、誤差を最小化する。

3. レイアウト結果にもとづくブロック位置の調整

下位階層のレイアウト結果に従い、フロアプランでのブロック位置を調整する。目標形状とレイアウト結果の差を

考慮して、配線領域が適切に確保されるようにブロック位置を決定する。

以下では、それぞれの処理について説明する。

3 目標形状の決定

フロアプランニングにおける未設計ブロックの面積予測には、いくつかの方法が提案されている [1][10][11][12]。しかし、フロアプランニングにおけるブロック配置・形状の決定時には、一般的に未設計ブロックの端子位置は未定であり、詳細なブロック間の配線領域は端子位置の決定後概略配線をおこなうことで見積もられる。この詳細な配線領域見積りを考えた場合、フロアプラン結果のブロック形状を変更した方がよい場合がある。例を図 2 で示す。このブ

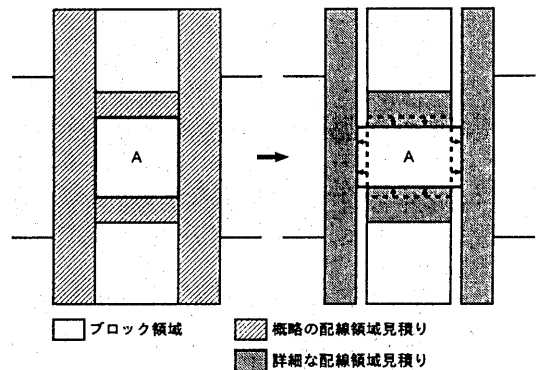


図 2: 目標形状の決定

ロック A が全体の高さに対して支配的である（チャンネルポジショングラフ [8] において垂直方向のクリティカルパス上に存在する）とする。このブロック上下の配線領域がフロアプラン時より大きくなり、ブロック左右の配線領域がフロアプラン時より小さい場合には、ブロック形状をフロアプランより横長にした方が全体の面積が削減される。

ここでは、配線領域の予測値を加えた形状関数の合成により、レイアウトに対するブロックの目標形状を決定する。[4]では、スライス構造における形状関数の合成方法が提案されている。ここで、各スライス木のノードに対応する形状関数を計算する際に、そのノードに対応する配線チャンネルの領域見積り値を加えて、配線領域を含めた形状関数を算出する。例えば、スライス木のノードが垂直分割を表す場合、配線領域は形状関数の幅に対して加えられる。スライス木の葉より上位階層の形状関数の算出を繰り返して全体の形状関数を算出した後、最上位ノードにおいて与えられた縦横比を満たしかつ面積の最小な形状を選択する。そして、最上位ノードより再帰的に各ブロックの形状を求める。この方法により、フロアプランの詳細化を反映したブロックの目標形状を決定する。

4 形状誤差の最小化

レイアウト結果の幅、高さともに目標形状より小さい場合には、上位階層に対する面積の影響はないが、幅、高さのいずれかが目標形状より大きい場合には、上位階層の面積増加を避けるように、レイアウト結果を求める。

まず、ブロック形状の変化が上位階層に影響しない範囲を、形状の余裕として定義する。次に、形状の余裕を用いてレイアウト結果の目標形状に対する誤差を定義する。そして、誤差が減少するようにレイアウト条件を変更して結果を改善し、誤差が最小となるレイアウト結果を求める。

4.1 ブロック形状の余裕

ブロック形状は、スライス木における上位ノードが占める領域に対して余裕を持つ場合がある。図3はその例である。ブロックAとBが横並びに配置されている場合を考える。配置に対応するスライス木が図

3の下の部分である（記号*はスライス構造での垂直分割を表している）。上位ノード*i*に該当する領域は点線で囲まれた部分である。ブロックBの高さがブロックAより高い場合には、ブロックAは上下方向に若干の余裕があることになる。しかし、水平方向を考えた場合にはブロックA,Bともに上位ノード*i*の幅を支配しており、AまたはBの幅の増加は上位ノード*i*の形状での幅の増加となる。そこで、スライス木の最上位ノードより再帰的に、各ノードの形状の余裕値を求める。

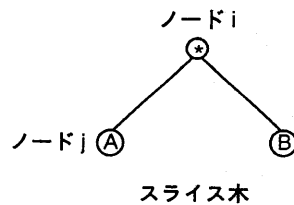
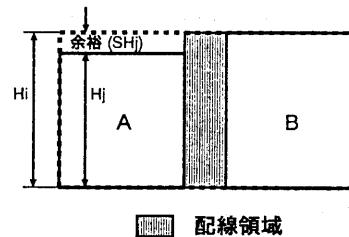


図3: ブロック形状の余裕

スライス木のノード*i*の占める幅、高さをそれぞれ W_i, H_i とし、ノード*i*の幅、高さの余裕値をそれぞれ SW_i, SH_i とする。同様に、ノード*j*の幅と高さを W_j, H_j とし、ノード*j*の幅と高さの余裕を SW_j, SH_j とする。

いま、スライス木のノード*j*がノード*i*の子であるとき、ノード*j*の余裕値を次のように定義する。

1. ノード i が垂直分割を表す時

$$SW_j = SW_i \times \frac{W_j}{W_i}$$

$$SH_j = SH_i + H_i - H_j$$

2. ノード i が水平分割を表す時

$$SW_j = SW_i + W_i - W_j$$

$$SH_j = SH_i \times \frac{H_j}{H_i}$$

スライス木の最上位ノード r に対して、 $SW_r = SH_r = 0$ とし、スライス木の最上位ノードより再帰的に各ノードの余裕値を算出する。垂直分割の場合、上位ノードの幅の余裕値は、下位ノードの幅の比にしたがって、下位ノードの幅の余裕値に分配される。

4.2 ブロック形状の誤差

ブロックの目標形状とレイアウト結果とのずれを誤差関数として次のように定義する。ブロック B に対応するスライス木のノードを b とし、目標形状の幅と高さをそれぞれ TW_b, TH_b とする。また、レイアウト結果の幅と高さをそれぞれ RW_b, RH_b とするとき、ノード b の幅の誤差 $E_w(b)$ と、高さの誤差 $E_h(b)$ を以下のように定義する。

$$E_w(b) = \begin{cases} \frac{RW_b - TW_b}{SW_b} & (RW_b - TW_b \geq 0) \\ \frac{1}{RW_b - TW_b} & (RW_b - TW_b < 0) \end{cases}$$

$$E_h(b) = \begin{cases} \frac{RH_b - TH_b}{SH_b} & (RH_b - TH_b \geq 0) \\ \frac{1}{RH_b - TH_b} & (RH_b - TH_b < 0) \end{cases}$$

ノード b の誤差 $E(b)$ は、

$$E(b) = \max(E_w(b), E_h(b))$$

と定義する。

レイアウト結果が目標を越えない場合には、必ず誤差関数の値は負となり、結果と目標との差が大きいほど誤差が大きくなる。結果が目標を越える場合には、必ず誤差関数の値は正となる。さらに誤差は、結果と目標との差を余裕値で割ることで定義しているので、幅方向と高さ方向の差が同じ場合には、余裕値が小さい方が誤差が大きくなる。

このようにして、上位階層に対する余裕を考慮した形状の誤差を定義している。

4.3 レイアウト結果の改善

下位階層がスタンダードセル方式でレイアウトされるブロックの場合には、セル行の数を変化させることで、形状の変更が可能である。目標形状とレイアウト結果との誤差 $E(b)$ を最小とするため、レイアウト結果の改善を次のように行う。

1. 目標形状となるセル行数を求める。
2. ブロックを自動レイアウトする。
3. 目標形状とレイアウト結果の誤差 $E_w(b), E_h(b), E(b)$ を求める。
4. $E_w(b) > E_h(b)$ のとき、現在のセル行数を増やし、予測形状を求め、予測誤差 $E'(b)$ を求める。
5. $E_w(b) < E_h(b)$ のとき、現在のセル行数を減らし、予測形状を求め、予測誤差 $E'(b)$ を求める。
6. $E_w(b) = E_h(b)$ のとき、ステップ 9へ進む。
7. $E'(b) < E(b)$ のとき、ステップ 2へ戻る。
8. $E'(b) \geq E(b)$ のとき、ステップ 9へ進む。
9. 最も誤差が少ない結果を選択し、出力する。

以上の処理を全ての下位ブロックに対し適用する。

セル行を変化させたときの形状予測には、レイアウト結果の情報を利用するため、非常に正確な予測が可能である。実験では、フロアプランニング時の面積予測精度が悪い場合でも、1ブロックあたりのレイアウト回数は2～3回で誤差最小の形状に収束する。

下位階層がマクロセル方式でレイアウトされるブロックの場合、その形状を変更を、目標形状の決定手法を再帰的に適用し実現する。これにより、階層的レイアウト工程は完全に自動化できる。階層構造が深いと最適化に時間がかかるが、現実的にはフロアプランニング段階で設計の複雑化を避ける（フロアプランニング設計も、タイミング検証も工数が増える）ため、レイアウト階層を編集し深い階層を防ぐことが多く、問題はない。

5 レイアウト結果にもとづくブロック位置の調整

下位階層のブロックのレイアウトが完了すると、最終的なブロック形状をもとに、ブロック位置の調整をおこなう。第3章と同様に、確定した各ブロックの形状関数を合成し、スライス木の各ノードの位置を求める。この結果、ブロック間には必要な配線領域が確保されるように、ブロックが配置される。

位置調整時も、目標形状決定時と同様に、スライス木における上位ノードの大きさに対して、下位ノードの大きさが余裕を持つ場合がある。その余裕をブロックの両側に均等に配分する。配分された余裕はブロック間配線の領域として使われ、配線領域の減少につながる。

調整されたブロック配置を用いて、マクロセル間配線を適用し、その階層のレイアウト工程が完了する。

6 本手法の評価

これまでに述べた手法を、当社のレイアウトシステムに組み込み、階層的レイアウト工程の自動化システムを試作した。この試作システムで、実際のLSIデータを用いて、本手法の評価を行った。

6.1 実験データ

二種類のデータを用いて本手法の評価を行った。二種類ともにレイアウト階層の深さは2階層で、最下層はスタンダードセル方式、または、既設計ブロックより構成され、上位階層はそれらのブロックをマクロセル方式でレイアウトする。

それぞれのデータについて、スタンダードセル方式および既設計方式のブロック数とそれぞれの面積和の比率を表に示す。

data		スタンダードセル方式	既設計方式
1	ブロック数	7	5
	面積比	1.45	1
2	ブロック数	19	14
	面積比	0.50	1

表 1: 実験データ

データ2の方がブロック数が多いが、スタンダードセル方式でレイアウトされる領域の比率は少ない。

6.2 実験結果

従来手法を1とした場合の面積の比を表に示す。ここで、従来手法は、下位ブロックの形状決定と、ブロック位置の調整を人手により対話的におこなった方法である。

表2のように、全体の面積、配線領域ともに、提案手法の方が従来手法より良い結果が得られた。特に、データ1はマクロセル間の配線領域が大きく減少しているが、これはスタンダードセル方式でレイアウト

データ 1

方式	チップ面積	マクロセル間 配線領域
従来手法	1	1
提案手法	0.963	0.862

データ 2

方式	チップ面積	マクロセル間 配線領域
従来手法	1	1
提案手法	0.968	0.929

表 2: 実験結果

する比率が高いため、本手法による最適化がより効果的に作用したと考えられる。

このように、実験により本手法は階層的レイアウト設計工程の完全な自動化が実現できる上に、人手による経験的な最適化と同等以上の品質が得られることが確かめられた。

7 まとめ

本論文では、階層的レイアウトで必要な最適化を行う手法を提案した。さらに、従来の自動レイアウト技術と結合して、階層的レイアウト工程を全自動化するシステムを試作した。本手法では階層的レイアウト工程において、これまで残されていた人手による最適化作業の自動化を実現し、生産性の向上を図ることができる。さらに、実際のレイアウトデータに適用した結果で実験評価をおこない、面積において良好な結果を得た。したがって、本手法によりレイアウト結果の品質は従来以上の結果が得られ、自動化のみならずLSIのコスト低減にも貢献することが確認できた。

参考文献

- [1] 山内、神戸: “階層構造の改善によるフロアプランニング最適化の一手法”, 第6回回路とシステム軽井沢ワークショップ, pp.67-72
- [2] 山田、中谷、吉岡、藤原、神戸: “高密度マクロセル間配線の一手法”, 信学技報 VLD91-87.
- [3] 吉岡、藤原、神戸: “マクロセル間配線におけるブロック位置最適化手法”, 信学全大1-97.
- [4] D.F.Wong and C.L.Liu: “A New Algorithm for Floorplan Design”, Proc.23rd D.A.Conf., pp.101-107,1986.
- [5] R.H.J.M.Otten: “Automatic Floorplan Design”, Proc.19th D.A.Conf., pp.261-267, 1982.
- [6] 山内、富田、松本、神戸: “VLSI フロアプランニングシステム”, 信学技報 VLD89-44.
- [7] W-M.Dai, T.Asano, and E.S.Kuh: “Routing Region Definition and Ordering Scheme for Building-Block Layout”, IEEE Trans. on CAD, vol.4, no.3, pp.189-196, 1985.
- [8] W-M.Dai, M.Sato, and E.S.Kuh: “A Dynamic and Efficient Representation of Building-Block Layout”, Proc.24th D.A.Conf., pp.376-384, 1987.
- [9] G.Zimmerman: “A New Area and Shape Function Estimation Technique for VLSI Layouts”, Proc.25th D.A.Conf., pp.60-65, 1988.
- [10] F.J.Kurdahi and A.C.Parker: “Techniques for Area Estimation of VLSI Layouts”, IEEE Trans. on CAD, vol.8, no.1, pp.81-92, 1989.
- [11] M.Pedram and B.Preas: “Interconnection Length Estimation for Optimized Standard Cell Layouts”, Proc.ICCAD, pp.390-393,1989.
- [12] T.Hamada, C-K.Cheng, and P.M.Chau: “A Wire Length Estimation Technique Utilizing Neighborhood Density Equations”, Proc.29th D.A.Conf., pp.57-61,1992