

Title	大規模集積回路のレイアウト合成手法に関する研究
Author(s)	福井, 正博
Citation	大阪大学, 1999, 博士論文
Version Type	VoR
URL	<a href="https://doi.org/10.11501/3155637">https://doi.org/10.11501/3155637</a>
rights	
Note	

*Osaka University Knowledge Archive : OUKA*

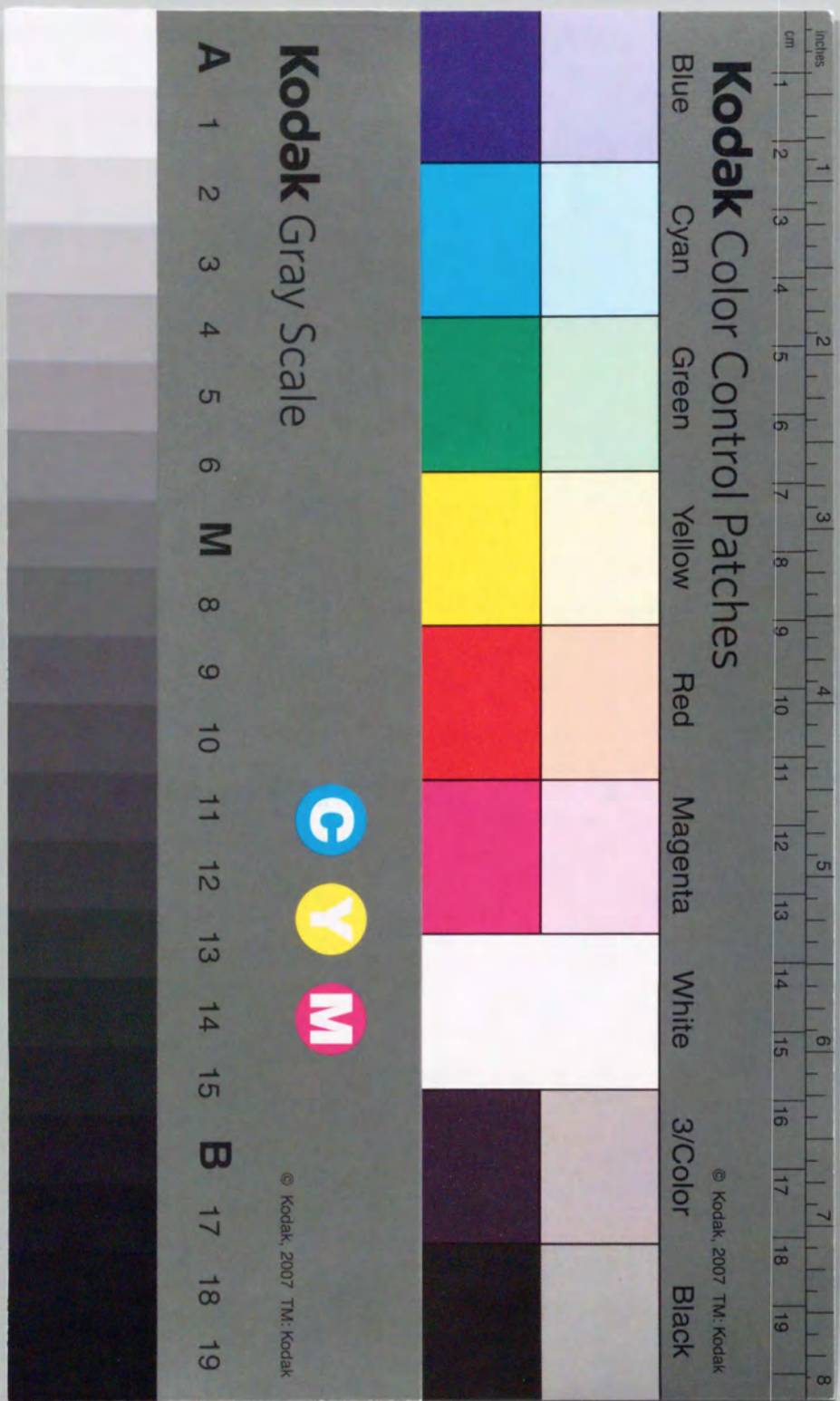
<https://ir.library.osaka-u.ac.jp/>

Osaka University

大規模集積回路のレイアウト合成手法  
に関する研究

1999年

福井 正博



①

大規模集積回路のレイアウト合成手法  
に関する研究

1999年

福井 正博

## 内容梗概

本論文は、著者が昭和56年から昭和58年にかけて大阪大学工学部電子工学科および大阪大学大学院工学研究科博士前期課程において、および、昭和59年から平成10年にかけて松下電器産業半導体研究センターおよび半導体先行開発センターにおいて、それぞれ行った大規模集積回路の配置配線手法に関する研究成果および標準セルの自動合成に関する研究成果をまとめたものである。

本論文は、大規模半導体集積回路のレイアウト設計の効率化と品質向上を論じるものであり、全7章から構成される。

本研究では、ゲートアレイ方式LSIの配置配線を行う場合に、セルの等価端子に対するネットの割り当てを最適化することによって、配線処理を簡易化できることに着目し、等価端子の割り当て手法を提案し、実験によりその有用性を明らかにする。

ついで、階層的にLSIの配置配線するためのチップフロアプラン法を考察し、複数の動作条件に対して電源幅を最適化する電源配線、および信号線の概略配線経路を最適化する方法を用い、配線の面積およびブロックの面積を精度良く推定することによってブロックの配置および形状を最適化する方法を提案し、実際のLSIチップへの適用例を示しながら、その実用性について述べる。

さらに、階層配置配線設計において、ブロックの設計が全て終わった後のブロック間配線を効率良く行う方法を考察し、面積の最小化とタイミングの最適化を同時に行う概略配線手法、ブロック間の位置を最適化し、ブロック間をL字形状を含むチャンネルに分割しグリッドフリーチャンネル配線を実行することにより、全ての詳細配線を実現する方法を提案し、その実用性について述べる。

最後に、ランダムロジックブロックや、データパスブロック、メモリ、等を構成するレイアウトの最小単位であるリーフセルのレイアウトの自動合成手法を考察し、トランジスタのサイズ決め、2次元のトランジスタ配置配線、コンパクションからなるリーフセルの合成手法を提案し、標準セルの合成への適用例を示し、その実用性を示す。

本論文は全体を7章に分けて構成する。

## 関連発表論文

### I. 学会誌等採録論文

1. M. Fukui, N. Shinomiya, S. Saika, T. Akino, and S. Kuninobu, "Layout abstraction and technology retargeting for leaf cells," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E81-A, No.12 (December 1998), to be published.
2. M. Fukui, M. Tanaka, and M. Imai, "Design optimization by using flexible pipeline modules," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E81-A, No.12 (December 1998), to be published.
3. T. Shiohara and M. Fukui, "A pin assignment and global routing algorithm for floorplanning," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E81-A, No.8 pp. 1725-1732 (August 1998).
4. S. Saika, M. Fukui, N. Shinomiya, and T. Akino, "A two-dimensional placement algorithm for cell synthesis and its application to standard cells," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E80-A, No.10, pp. 1883-1891 (October 1997).
5. M. Fukui, A. Yamamoto, R. Yamaguchi, and S. Hayama, "SMILE- Hierarchical layout system for building block LSI," *International Journal of Computer Aided VLSI Design*, Vol. 1, No. 3, pp. 281-302 (March 1989).
6. M. Fukui, A. Yamamoto, R. Yamaguchi, S. Hayama, and Y. Mano, "A block interconnection algorithm for hierarchical layout system," *IEEE Transactions on Computer-Aided Design of Integrated Circuits And Systems*, Vol. CAD-6, No.3, pp. 383-391 (May 1987).
7. 福井正博、築山修治、白川功、"ゲートアレイ方式 LSI の端子割り当て問題に対する一手法"、電子情報通信学会論文誌(C), Vol. J66-C, No. 12, pp. 1172-1179 (1983年12月).
8. S. Tsukiyama, I. Harada, M. Fukui, and I. Shirakawa, "A new global router for gate array LSI," *IEEE Transactions on Computer-Aided Design of Integrated Circuits And Systems*, Vol. CAD-2, No. 4, pp. 313-321 (October 1983).

### II. 研究集会等発表論文(査読あり)

1. M. Tanaka, M. Fukui, and S. Kuninobu, "A precise estimation model of cell area and transistor capacitance for transistor size optimization," in *Proc. Synthesis and System Integration of Mixed Technologies*, pp. 241-247 (October 1998).
2. 田中正和、福井正博、雑賀俊二、"正確な面積/容量推定にもとづくトランジスタサイズ最適化手法"、電子情報通信学会回路とシステム軽井沢ワークショップ, pp.125-130 (1998年4月).
3. M. Fukui, M. Tanaka, and M. Imai, "Design optimization by using flexible pipelined modules," in *Proc. Synthesis and System Integration of Mixed Technologies*, pp. 176-183 (December 1997).
4. S. Saika, M. Fukui, N. Shinomiya, and T. Akino, "A two-dimensional placement for cell synthesis," in *Proc. on Asia and South Pacific Design Automation Conference*, pp. 557-562

(January 1997).

5. 福井正博、四宮典子、雑賀俊二、秋濃俊郎、"テクノロジフリーレイアウト合成方法"、電子情報通信学会回路とシステム軽井沢ワークショップ, pp. 229-234 (1996年4月).
6. T. Akino, M. Fukui, T. Sawai, and N. Shinomiya, "Relations between logic synthesis and physical synthesis in deep sub-microns," in *Proc. International Workshop on IP Based Synthesis and System Design*, pp. 33-39 (December 1995).
7. M. Fukui, N. Shinomiya, and T. Akino, "A new layout synthesis for leaf cell design," in *Proc. Asia and South Pacific Design Automation Conference*, pp. 259-264, (August 1995).
8. N. Shinomiya, M. Fukui, and T. Akino, "A new sea-of-cells style layout synthesis," in *Proc. Synthesis and System Integration of Mixed Technologies*, pp. 17-23 (July 1995).
9. M. Fukui, and A. R. Newton, "Optimum module generation for semi-custom," in *Proc. IEEE Asia-Pacific Conference on Circuits and Systems*, pp. 184-189 (December 1992).
10. M. Fukui and A. R. Newton, "Multi-output module generation for semi-custom design," in *Proc. Synthesis and Simulation Meeting and International Interchange*, pp. 181-190 (June 1991).
11. M. Fukui, Y. Tanaka, and T. Akino, "An algorithm for power and ground routing in building block VLSI," in *Proc. Synthesis and Simulation Meeting and International Interchange*, pp. 294-299 (January 1990).
12. M. Fukui, A. Yamamoto, R. Yamaguchi, S. Hayama, and Y. Mano, "A new routing algorithm for building block layout system," in *Proc. European Conference on Circuit Theory and Design*, pp. 447-452 (September 1987).
13. M. Fukui, S. Hayama, and Y. Mano "A new block interconnection algorithm for VLSI layout system," in *Proc. International Symposium on Circuits and Systems*, pp. 1039-1042 (June 1985).
14. S. Tsukiyama, M. Fukui, and I. Shirakawa, "A heuristic algorithm for a pin assignment problem of gate array LSI's," in *Proc. International Symposium on Circuits and Systems*, pp. 465-469 (May 1984).
15. S. Tsukiyama, I. Harada, M. Fukui, and I. Shirakawa, "A placement and routing algorithms for gate array LSI," in *Proc. International Conference on Computer Design: VLSI in Computers*, pp. 596-599 (October 1983).
16. S. Tsukiyama, I. Harada, M. Fukui, I. Shirakawa, and H. Ozaki, "An algorithm of global routing for master slice LSI," in *Proc. International Symposium on Circuits and Systems*, pp. 1009-1012 (May 1982).

### III. その他研究会等発表論文

1. 四宮典子、福井正博、西垣泰男、"リーフセル用配線システム," 信学技報, CPSY97-71, pp. 21-26. (1997年10月).
2. 田中正和、福井正博、"レイアウトを考慮したトランジスタサイジングの一手法," 信学技報, VLD96-100, pp. 31-38 (1997年3月).
3. 四宮典子、福井正博、田中正和、雑賀俊二、秋濃俊郎、"屈曲ゲートを用いたレイアウト最適化手法とその評価," 電子情報通信学会ソサイエティ大会, p. 59 (1996年8月)

4. 雑賀俊二、福井正博、四宮典子、秋濃俊郎、”セル合成におけるトランジスタ配置手法”、信学技報、VLD95-156, pp. 105-112 (1996年3月).
5. 四宮典子、福井正博、雑賀俊二、秋濃俊郎、”屈曲ゲートを用いたレイアウト最適化手法”、信学技報、VLD95-155, pp. 97-104 (1996年3月).
6. 福井正博、雑賀俊二、秋濃俊郎、”トランジスタのレイアウトモデルに柔軟性を持たせたコンパクト化方法”、信学技報、VLD95-50, pp. 29-37 (1995年6月).
7. 四宮典子、福井正博、秋濃俊郎、”シー・オブ・セルズ型レイアウト・アーキテクチャの提案”、信学技報、VLD94-115, pp. 45-52 (1995年3月).
8. 福井正博、秋濃俊郎、”VLSI のリーフセル合成に関する一手法”、信学技報、VLD94-65, pp. 13-19 (1994年10月).
9. 川上善之、山口龍一、福井正博、羽山繁、秋濃俊郎、”論理回路の規則的繰り返しが多く存在するブロックにおけるセル配置の一手法”、信学技報、VLD89-48 pp. 73-79 (1989年10月).
10. 塩原孝弘、福井正博、羽山繁、”フロアプランにおけるピン配置の一手法”、信学技報、VLD88-87, pp. 9-16 (1989年2月).
11. 福井正博、岩崎知恵、羽山繁、”チャンネル構造列挙の一手法”、情処研報、DA44-10, pp. 75-82 (1988年10月).
12. 福井正博、山本敦志、岩崎知恵、羽山繁、”ブロック形状最適化の一手法”、電子情報通信学会秋季全国大会, pp. A-1-163~164 (1988年9月).
13. 福井正博、山本敦志、岩崎知恵、羽山繁、”VLSIレイアウト設計におけるブロック形状ピン配置の最適化の一手法”、信学技報、VLD88-4, pp. 25-32 (1988年4月).
14. 重本一郎、福井正博、羽山繁、”自動レイアウトシステムにおけるレイアウト検証の手法”、情報処理学会第36回全国大会, pp. 2027-2028 (1988年3月).
15. 福井正博、塩原孝弘、高木善之、羽山繁、”自動レイアウトシステムにおける遅延時間最適化の手法”、情報処理学会第36回全国大会, pp. 2025-2026 (1988年3月).
16. 山口龍一、福井正博、山本敦志、羽山繁、”階層的レイアウトシステムにおけるスタンダードセル自動配線の一手法”、情報処理学会第34回全国大会, pp. 1969-1970 (1987年10月).
17. 山本敦志、福井正博、山口龍一、羽山繁、”階層的レイアウトシステムにおけるスタンダードセル自動配置の一手法”、情報処理学会第34回全国大会, pp. 1991-1992 (1987年10月).
18. 山本敦志、福井正博、羽山繁、間野洋治郎、”シミュレティッドアニーリング法を用いたスタンダードセル自動配置の一手法”、信学技報、CAS86-191, pp. 25-31 (1987年1月).
19. M. Fukui, A. Yamamoto, R. Yamaguchi, S. Hayama, and Y. Mano, ”Hierarchical routing system for building block LSI,” 信学技報, CAS87-66, pp. 99-104 (July 1987).
20. 福井正博、山本敦志、山口龍一、羽山繁、間野洋治郎、”階層的レイアウトシステムにおける自動配線の一手法”、信学技報、CAS85-149, pp. 9-16 (1986年1月).
21. 福井正博、山本敦志、羽山繁、間野洋治郎、”チャンネル構成の一手法”、電子情報通信学会半導体・材料部門全国大会, pp. 70-71 (1985年10月).
22. 福井正博、竹田信弘、築山修治、白川功、尾崎弘、”マスタスライス方式LSIに対する一配線手法”、信学技報、CAS82-148, pp. 55-60 (1982年2月).
23. 福井正博、築山修治、白川功、尾崎弘、”チャンネル配線の一手法”、信学技報、CAS81-121, pp. 91-98 (1981年2月).
24. 福井正博、築山修治、白川功、尾崎弘、”チャンネル配線の一手法”、電子情報通信学会全国大会, p. 2-336 (1980年8月).

## 目次

第1章 序論 .....	1
第2章 大規模集積回路設計とレイアウト合成技術の概要 .....	2
2.1 緒言 .....	2
2.2 ゲートアレー方式LSIのレイアウト設計の流れとその課題 .....	5
2.3 階層的レイアウト設計の流れとその課題 .....	5
2.3.1 チップフロアプランにおける課題 .....	5
2.3.2 ブロックレイアウトにおける課題 .....	5
2.3.3 ブロック間配線における課題 .....	6
2.4 セルレイアウト設計の流れとその課題 .....	6
2.5 本研究の位置付けと意義 .....	7
2.6 結言 .....	8
第3章 ゲートアレー方式LSIの等価端子割り当て手法 .....	9
3.1 緒言 .....	9
3.2 等価端子割り当て問題の定義 .....	10
3.3 等価端子割り当てのアルゴリズム .....	12
3.4 実験と結果の考察 .....	17
3.5 結言 .....	19
第4章 階層的配置配線におけるフロアプラン手法 .....	20
4.1 緒言 .....	20
4.2 チップフロアプラン問題 .....	20
4.2.1 レイアウトモデル .....	20
4.2.2 階層的配置配線フロー .....	22
4.3 ブロック間配線とピン配置 .....	24
4.3.1 初期配置アルゴリズム .....	25
4.3.2 分散配置アルゴリズム .....	26
4.4 電源配線 .....	27
4.4.1 電源配線問題 .....	28
4.4.2 電源配線アルゴリズム .....	29
4.5 ブロック形状最適化 .....	30
4.6 ブロック面積の推定 .....	33
4.7 フロアプラン実験例 .....	33
4.8 結言 .....	34
第5章 ブロック間配線手法 .....	35
5.1 緒言 .....	35
5.2 配線構造の定義 .....	35
5.3 概略配線 .....	36
5.3.1 処理概要 .....	36

5.3.2 初期配線	36
5.3.3 再配線	37
5.4 詳細配線	38
5.4.1 処理概要	38
5.4.2 ブロックの位置決め	38
5.4.3 チャンネル構造決め	39
5.4.4 チャンネル配線	40
5.5 実験と結果の考察	42
5.6 結言	44
第6章 標準セルの自動合成手法	45
6.1 緒言	45
6.2 テクノロジマイグレーション処理フロー	47
6.3 レイアウト抽象化モデル	47
6.3.1 配置要素の抽象化モデル	47
6.3.2 配線の抽象化モデル	48
6.4 レイアウト再合成処理概要	48
6.5 アルゴリズム	50
6.6 実験と結果の考察	53
6.7 結言	54
第7章 結論	56
謝辞	58
参考文献	59

## 第1章 序論

近年、大規模集積回路の高集積化および高性能化の要望はとどまるところを知らず、システムオンチップと言われるように、1チップ上にメモリやアナログの入出力、プロセッサ等を同時に搭載されるようになってきている。このような設計規模の増大に対応して、設計効率の向上が重要な課題となっている。一方、半導体集積回路の微細化技術は、図 1.1 に示すように、2000年には  $0.18\mu\text{m}$  の時代を迎え、配線遅延の増大や、製造ばらつきの増大、電源電圧降下やノイズの増大など、設計時に考慮すべき問題も複雑化している。[1]

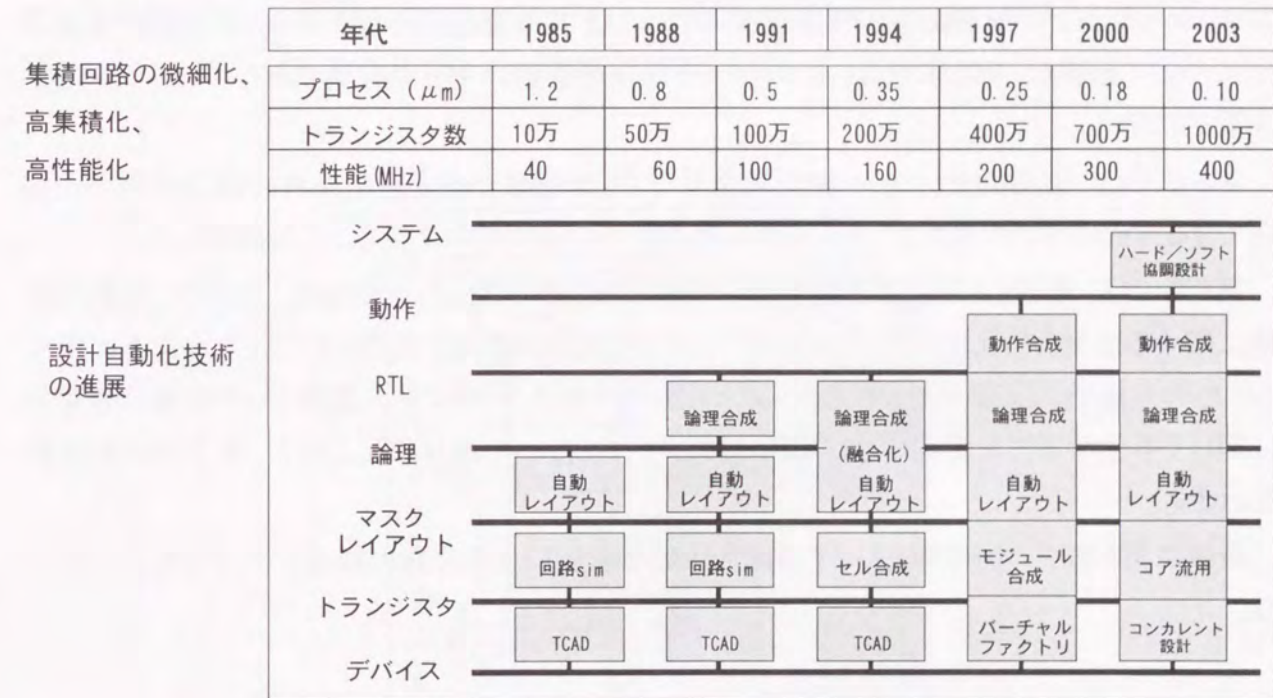


図 1.1 大規模集積回路の規模の増大と設計自動化の変遷

設計自動化の流れを遡ると、1970年代には、標準セル方式の設計手法が提案された。これは、NAND やフリップフロップ等の論理要素を標準セルという形で準備し、それらをチップ上に配置配線する方式である。論理レベルで回路およびレイアウト設計を行うことが可能となったことと、標準セル自動配置配線技術との相乗効果によって、回路レイアウト設計効率が大幅に向上した。1980年代には、論理回路の自動合成技術が実用化され、論理回路設計工数も大幅に向上した。その後、機能記述からレイアウトを合成するシリコンコンパイラや、規則性の高いレイアウトを自動化するモジュール合成システムなども提案され、大規模集積回路の設計自動化技術は産業の発展に多大な貢献をしてきた。

1990年代のディープサブミクロン時代を迎えるころから、信号遅延に占める配線遅延の比

率が増加傾向を示すようになってきた。しかも、設計規模の増大比率は依然大きく、効率的でかつ物理的条件を考慮したレイアウト算法を深く追及すべき状況にある。本論文は、このような背景において、主に物理レベル合成・最適化について著者が行ってきた研究をまとめる。

第2章では、用途によって異なる大規模集積回路のレイアウト構造と方式、各方式におけるレイアウト合成の課題について概説した後、本研究の位置付けと、意義目的について述べる。

第3章では、ゲートアレイ方式LSIの設計TAT(turn around time)を短縮するために有効な等価端子割当手法について提案を行い、その有効性を実験によって確める。

第4章では、多くの機能ブロックから構成されるシステムLSI に対するレイアウト合成のためのチップフロアプラン手法の提案を行う。また、フロアプランの部分問題へのアプローチとして、ブロックの外部ピン配置および概略配線手法、および電源配線手法、ブロック形状最適化手法について提案し、第2章で示したレイアウト合成の課題に対して有効であることを、理論的、実験的に示す。

第5章では、実用的なブロック間配線に対して、面積最小化と遅延条件の満足の両方を実現できる方法について提案し、いくつかの実 LSI への適用例によって、その効果を示す。

第6章では、最近、とみに重要性を増してきたライブラリセルの自動生成、および、再利用技術に関する提案を行う。特に、プロセステクノロジーが更新されたときにレイアウトセルを再利用するための方法として、従来のコンパクション手法では困難とされていた配線ジョグ位置の最適化に関して有効な方法を提案する。実際のライブラリ設計への適用例によって、本手法の有効性を示す。

最後に第7章で、本研究によって達成された内容をまとめ、今後の課題に対する展望を述べる。

## 第2章 大規模集積回路設計とレイアウト合成技術の概要

### 2.1 緒言

一般に大規模集積回路はその開発方法により、大きく、セミカスタムLSIと、カスタムLSIの2種類に分類される。セミカスタムLSIは、スタンダードセル方式、ゲートアレイ方式等に代表され、用途としては多品種少量生産向きである。ゲートアレイ方式は、あらかじめトランジスタをアレイ状に配置し、拡散工程までを済ませたマスタースライスを準備しておき、受注に応じて金属配線以降の工程をす方式である。拡散工程までが事前に行われているので、受注から出荷までの製造期間が短い。ゲートアレイ方式LSIのレイアウト構造の例を図 2.1 に示す。アレイ状に配置されたセル行間に配線を施すためのチャンネル領域を設ける。セル上およびチャンネル上に配線を施すことによって、論理機能を持つ回路が実現される。

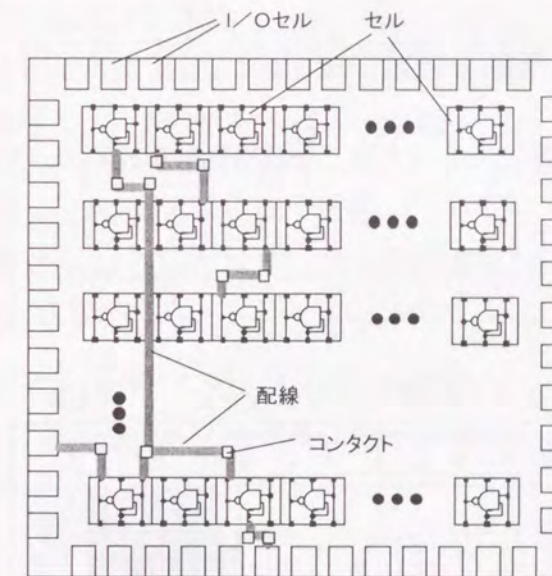


図 2.1 ゲートアレイ方式 LSI のレイアウト構造例

スタンダードセル方式のレイアウト構造を図 2.2 に示す。スタンダードセル(標準セル)と言われる論理機能を有するセルをあらかじめライブラリとして準備しておき、これらのセルを LSI チップ上に配置配線することによって、レイアウトを実現する方式である。同方式は、レイアウト設計後、製造工程を実施するので、配線領域をあらかじめとっておく必要は無く、必要最小限のチャンネル領域を確保すれば良い。ゲートアレイ方式に比べて製造工程に時間がかかるが、セルのレイアウトや配線領域が最小化できるため面積効率が良い。ゲートアレイよりは多量生産向きである。



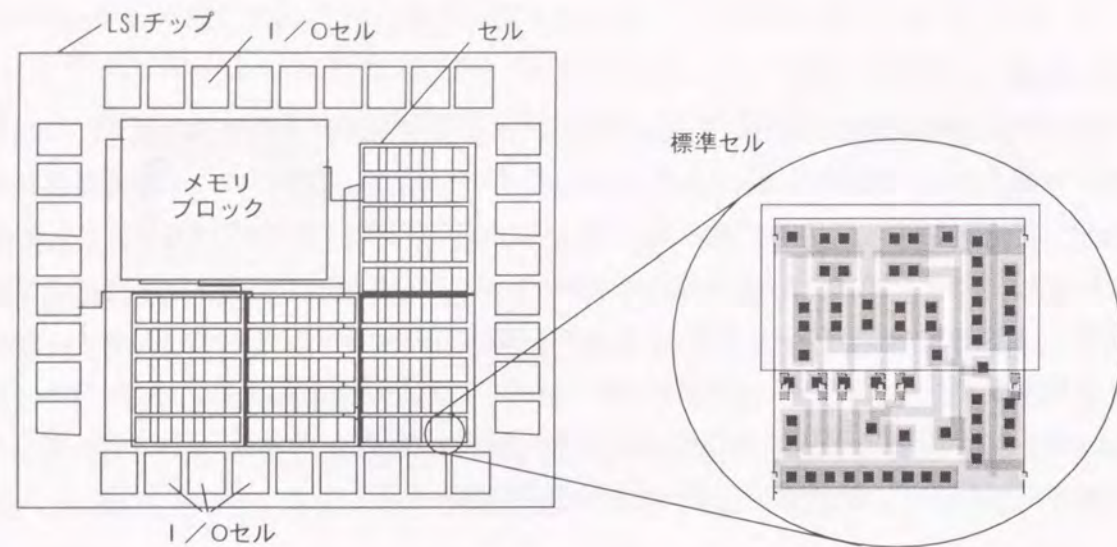


図 2.2 スタandardセル方式 LSI の例

カスタム方式は、特に多量生産向き LSI の設計に適用されるものであり、面積や遅延時間を最大限に抑える方式である。同方式は、開発工数を抑えるために、標準セルによって構成するブロックも含むが、多くはカスタム設計によって得られるブロックから構成される。同方式は比較的大規模なものが多く、図 2.3 に示すようなビルディングブロック方式設計が持ちいられる。

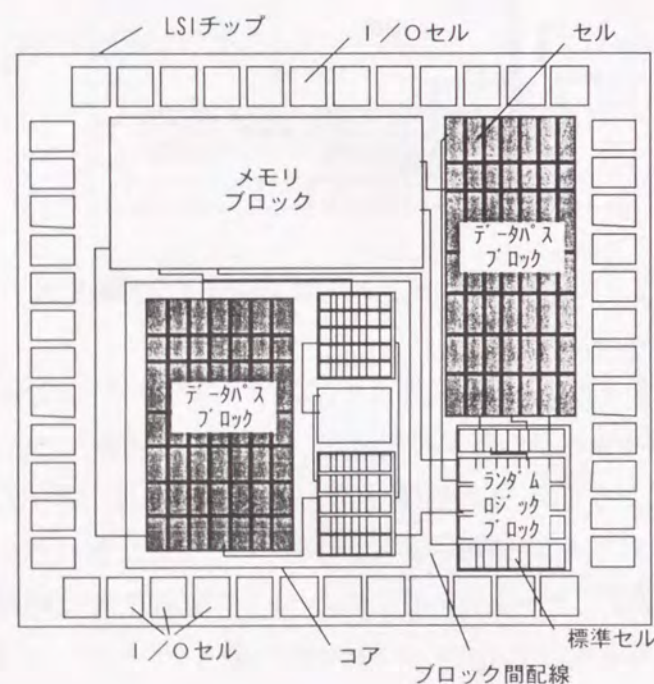


図 2.3 カスタム LSI の例

## 2. 2 ゲートアレイ方式 LSI のレイアウト設計の流れとその課題

ゲートアレイ方式 LSI では、配線のためのトラック数があらかじめ定まっている。レイアウト工程は、配置、概略配線、詳細配線の順に行われるが、詳細配線の段階で配線が不可能とわかれば、設計修正が生じ、大幅な設計コスト増につながる。このような不利益を回避するために、設計のできるだけ初期段階で後段の工程を単純化する処理が求められる。

## 2. 3 階層的レイアウト設計フローとその課題

前述の図 2.3 に示したビルディングブロック方式は、各々の機能ブロックを設計した後、それらをブロック間配線によって組み上げるという階層的なレイアウト方法が用いられる。同方式のレイアウト設計は、概ね、(1)チップフロアプラン、(2)ブロックレイアウト、(3)ブロック間配線の3段階から構成される。本節ではそれぞれの処理内容と課題について述べる。

### 2. 3. 1 チップフロアプランにおける課題

ブロックの配置とブロック形状の最適化、ブロック外部ピン配置と概略配線、電源配線経路等を決定する処理である。実用上、以下の点を考慮する必要がある。

- (i) ブロックの概略配置は人手で決める場合が多い。その場合には、できるだけ指定された配置を尊重して、ブロックの形状最適化を行う。
- (ii) フロアプラン設計は、形状可変なブロックに対して形状に対する要求を与え、各々ブロックが同要求に近いサイズで設計されること想定している。形状に対する要求と実際に設計されるブロックの形状や面積との間にずれがあったり、ブロック間配線領域の推定精度が低い場合には、無効領域やチップ面積増の原因となる。
- (iii) ブロック外部ピン配置およびブロック間概略配線は、ブロックおよびチップ面積、ブロック間配線長に与える影響がそれぞれ大きい。ブロック間の面積がチップ全体の約4~6割程度を占めることや、ブロック間配線遅延がシステム速度に大きな影響を与えるなどの要因を考えると、ブロック間配線の最適化がブロック面積最小化以上に重要である。しかしながら、外部ピンが局所的に集中して配置されると、ブロック内およびブロック間の一部に配線が集中し面積増をもたらすので、外部ピンをある程度分散させることも重要である。
- (iv) 電源配線に関しては、各々のブロックの消費電力は動作状態に応じて常に変化していることを考慮する必要がある。どの状態でもエレクトロ・マイグレーションや熱破壊が起きないように配線幅を確保し、ブロック電源端子の電圧低下を抑え、しかも、チップ面積を最小化することが重要である。従来手法では、各々のブロックの最大消費電流や、平均消費電流に基づき電源幅を求めるものは存在したが、必要最小幅にならず面積の増大を招いたり、すべての状態に対して障害が起きないことを保証できないなど課題があった。

### 2. 3. 2 ブロックレイアウトにおける課題

標準セル方式ブロックの配置配線に対しては、タイミングを満足し、かつ、面積を最小化する

配置配線が求められる。[2,3] モジュール合成に関しては、メモリ等のセル2次元アレイを合成するのに適したアレイコンパイラ[4]や、データバス等には、セル合成を用いてトランジスタレベルからブロックを合成する方法[5]などが用いられる。いずれに対しても人手設計並みの高集積度と、設計期間の短縮が求められる。

### 2.3.3 ブロック間配線における課題

ブロック間配線は、レイアウトが完了したブロック間の配線を行う問題である。同問題に対して提案されている方法は、大きく2種類に分割される。1つは線分探索または迷路法を用いてチップ全体を同時に配線する手法[6,7]であり、もう一つはブロック間配線領域を隣接するブロック間の帯状の配線領域(チャンネル)に分割し、各チャンネル毎に配線を行う手法である。[8,9]

線分探索を手法は、ブロック上の空き領域を自由に通過でき、多層化への展開が容易であるなどの柔軟な配線ができる反面、あらかじめブロック位置を固定してから配線を行うため未配線や無効領域が生じやすい。また、チャンネル配線手法より同時に取り扱う領域が広いと、計算時間および記憶容量が大となる。一方、チャンネル配線手法は、チャンネル構造が決まれば、高速でかつ未配線を生じずに配線を行なえるため、比較的大規模なチップに対しても効率的にレイアウトできるメリットがある。

ブロック間配線はブロック内以上に、チップ面積や遅延に及ぼす影響が大きいため、タイミングを考慮したチップ面積最小化手法が強く求められる。

### 2.4 セルレイアウト設計の流れとその課題

標準セルや I/O セル等のトランジスタレベルのレイアウト設計は、高集積化が要求されることから、従来は人手設計が中心であった。しかし、近年のプロセス設計ルールの複雑化、設計量の増大に伴い、設計自動化の要望が高まっている。[10]

セル設計は、新規に回路からレイアウト設計までを通して行う場合もあるが、設計期間の大幅な短縮要望により、古いプロセスで設計した回路やレイアウト設計を流用し、新しいプロセスにあったレイアウト設計を行うテクノロジマイグレーションの重要性がより高まっている。

回路からレイアウトまでの設計フローとそれぞれの課題を以下に示す。

- (1) 回路設計: 動作記述からトランジスタレベル回路を設計する処理である。2進関数のグラフ的表現方法であるBDD(Binary Decision Diagram)からトランジスタのスイッチングネットワークを合成する方法[11,12]が提案されており、パストラジスタロジックに対する実用化が進んでいる。しかしながら、一般のトランジスタ回路を合成する技術に関して十分有効な手法は提案されていない。
- (2) トランジスタサイジング: 回路構造が決まった後に、各々のトランジスタサイズを最適化してやることによって、所望の性能の回路を得る処理である。[13] トランジスタの性能は、ゲート幅だけでなく、拡散やゲート、配線等の寄生容量に大きく依存するため、それらを

考慮して回路最適化を行うことが重要である。[14]

- (3) トランジスタ配置: トランジスタの配置を求める処理である。1985年に上原他[15]による1次元トランジスタ配置のセル合成が提案されて以来、数多くのシステムや手法が提案されている。[16] しかし、人手設計されるセルは1次元のみならず、2次元的な配置によって面積削減を図っている。自動化に於いても、2次元配置モデルを用いて人手設計の面積以下にすることが求められる。[17]
- (4) トランジスタ配線: トランジスタ間の配線を行う処理である。[18,19] セル内のトランジスタ間配線においては、セルの高さ制約の考慮や、トランジスタからの配線引き出し位置決め等の問題がある。[19]
- (5) コンパクション: トランジスタやコンタクト等の配置要素と配置要素間の配線を含む図形をデザインルールを満たし、かつ、最小面積となるようにコンパクションする処理である。最近、プロセス設計ルールの複雑化により、プロセス開発からデザインルール決定までの期間が長期化する傾向にある。また、デザインルールが一旦決定しても、後の小変更が多発している。このため、コンパクション処理の重要性が増している。従来法の中では、縦あるいは横のいずれかの方向のみを扱う1次元コンパクションより、それらを同時に扱う2次元コンパクションが効果的である。[20,21] しかし、配置要素間の配線のジョグ発生およびコンパクションを効果的に扱うことが、困難な問題として残っている。[20]
- (6) 最終レイアウトからの回路・遅延パラメータの抽出とライブラリ登録。

テクノロジマイグレーションにおいては、上記の(2)および(5)の処理が必要であり、課題としても同様の課題が存在する。

### 2.5 本研究の位置付けと意義

以上で述べたように大規模集積回路のレイアウト合成問題は複雑多岐に渡り、しかもその規模が増している。効率良く良好な解を得るためのレイアウト合成技術の研究は、今後も益々重要となることが予想される。本研究は、このような大規模化・複雑化する大規模集積回路のレイアウト合成問題について検討を加え、新しい有用な手法を構築することを目的とする。

本研究の第1の意義は、ゲートアレイ方式LSIにおいて、配線成功率を高め、手戻りの少ない設計フローを構築し、設計TATの短縮を図ることである。ゲートアレイ方式LSIのレイアウト設計において、チャンネル配線を容易化するための等価端子割当手法について述べる。本手法の特徴は、チャンネル配線時に問題を複雑化する主要因である上下制約をできるだけ取り除いてやることであり、実験により配線率が向上することを示す。[22]

本研究の第2の意義は、大規模化するフルカスタム方式LSIの設計工数と期間を短縮すると同時に、面積・遅延において良好な結果を得ることである。本論文では、ブロックの形状や、ブロック間配線が占める面積に対する推定精度を向上させ、チップフロアプラン時の推定値と、ブロックレイアウトおよびブロック間配線の実現後の結果とのずれを極力減らし、無駄な領域の発

生をできるだけ抑える手法を提案する。[23] 同処理フローに関しては、ブロックの配置と形状最適化のそれぞれの処理を分離する方法を提案する。与えられたブロック配置の隣接配置関係を崩さない範囲でチャンネル構造を列挙し、それらの中から最も良好な解を選択する。更に、チップフロアプランの部分問題として、チップ面積の最小化とブロック間配線長の最小化を目標としてブロック外部ピン位置の最適化を行うピン配置手法の提案[24]と、動的な動作条件を考慮に入れた電源配線手法を提案する。[25] 次に、ブロック間配線問題に対しては、タイミング制約を満足しつつ配線面積を最小化する概略配線方法、ブロック間の配線を複数の直線型およびL字型チャンネルに分割し、全ての配線チャンネルの処理順を決定する方法、ブロックの相対距離を最適化し配線後のチップ面積を最小化する方法等についてそれぞれ提案し効果を示す。[26]

本研究の第3の意義は、プロセス変更や設計変更に対応できるレイアウト設計再利用技術を構築することである。トランジスタレベルからのレイアウト合成の要素技術として用いられるレイアウト再生成手法について提案する。従来のレイアウト再生成技術として良く用いられるコンパクション方法では、配線ジョグの発生をどこにどれだけ設けるかを決定することが困難な問題として知られていたが、本手法では配線情報を一旦抽象化した後、図形の移動と同時に配線を再生成する方法を用いる。実際のライブラリセル設計に用いて良好な結果を得ることを示す。[27]

## 2.6 結言

本章では、大規模集積回路のレイアウト設計について概要を示しレイアウト合成技術の直面する課題、および、本研究の位置づけと意義について述べた。

ゲートアレイ方式 LSI は、配線領域が固定されるため、配線の成功率を高めることが全体の設計期間短縮において重要である。本研究では、等価端子割り当て手法と有効性を示すことを第1の意義とする。

次に、大規模なカスタム方式 LSI に対する階層的レイアウト設計に対しては、高精度なブロックおよびブロック間配線の面積推定に基づくチップフロアプランの最適化が重要である。また、ブロック間配線は、未配線を無くすことによりレイアウト設計を効率化すると共に、タイミング制約を満たす面積最小の解を得ることが重要である。本研究においては、チップフロアプランとブロック間配線に適した処理フローと最適化手法を提案し、大規模化するフルカスタム方式LSIの設計工数削減し、かつ、面積・遅延において良好な結果を得ることを第2の意義とする。

最後に、標準セル設計に関しては、最近のプロセス設計ルールの複雑化の影響により、テクノロジマッピング、特に、コンパクション自動化の要望が増している。本研究においては、従来困難であった配置要素間の配線のジョグ発生および配置要素の移動を同時に扱う問題に対する手法提案を行い、その有効性を示すことを第3の意義とする。

## 第3章 ゲートアレイ方式 LSI の等価端子割り当て手法

### 3.1 緒言

集積回路技術の進歩に伴い、少量多品種の LSI の生産量が増大しているが、このような LSI を効率良く生産するために、ゲートアレイ方式 LSI が良く用いられ、又そのレイアウト設計には自動配置配線設計システムが運用されている。[28,29] このゲートアレイ方式 LSI においては、配線領域があらかじめ固定されているため、配線が一部のチャンネルに集中せずチップ全体に均一に分布することが、100%の配線率を得る上で重要となる。このため、配線設計は通常各ネットのチャンネル単位の大まかな配線経路を求めるグローバル配線、次に各ゲートセルが持つ論理的に等価な端子のうちのどれに各ネットの信号線を接続すべきかを決定し、チップ全体の配線問題をいくつかの2行2層配線問題に分割する端子割り当て、最後に各チャンネルでの詳細な配線を決定するチャンネル配線の3つの段階に分けて行われる。このうち端子割り当てに対しては、それが次のチャンネル配線の成否に大きな影響が知られているが、これに対する有効な手法はあまり報告されていない。

本章では、このシステムにおいて用いられている端子割り当て手法について考察する。本手法は、各チャンネルにおける上下制約グラフ[30,31]上のサイクルの個数および配線密度を減少させることを目的として、全てのチャンネルを総合的に考慮しつつ端子割り当てを行うものである。本手法の効果を調べるために、実際の回路例に対して、乱数を用いて端子割り当てをおこなった場合との比較もおこなったので、それについても報告する。

### 3.2 等価端子割り当て問題の定義

本論文で対象とするゲートアレイ方式LSIのチップ構造を図3.1に示す。各セルは一つの論理ゲートの機能を持ち、同図に示すように、所下の辺上に電氣的に等価な端子を持つ。1/Oセルに囲まれたセル以外の部分が配線領域で、これは水平及び垂直チャンネルに分けられる。各垂直チャンネルにおいて、一つのセル行上の隣接する2つのセルに挟まれた部分を貫通セル(feedthrough cell)と呼ぶ。

グローバル配線は、与えられた各ネットの配線経路がどのチャンネルを通過するかを決定するものであるが、(例えば図3.2(a)参照)、この配線経路に対して、端子割当問題を記述するために、次のような信号ピンおよびその接続要求を定義する。すなわち、各ネットの配線経路がセルに接続する部分及び貫通セルを通過する部分のそれぞれに対応して信号ピンを設け、これらに付随して、各チャンネルでの配線経路の方向を接続要求として表わす。例えば、図3.2(a)のような配線経路に対しては同図(b)のような  $P_i$  から  $P_j$  の信号ピン、及び矢印で示されるような各信号ピンの接続要求が与えられる。

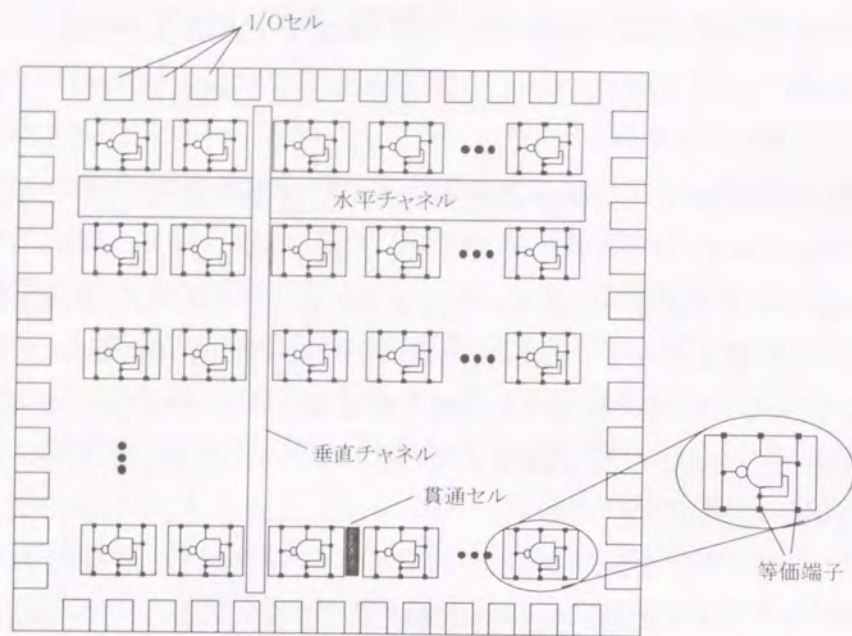


図 3.1 ゲートアレイ方式 LSI のレイアウト構造

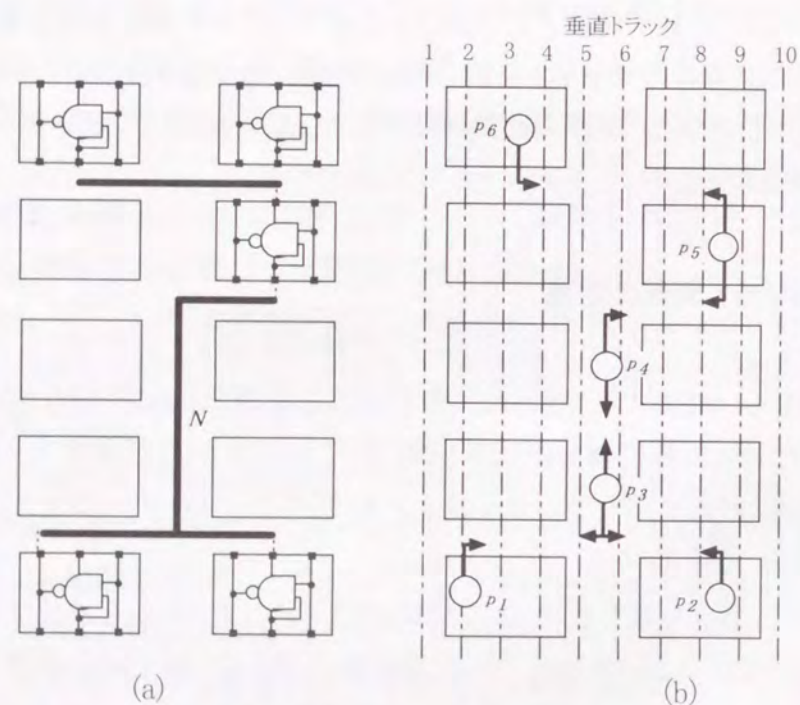


図 3.2 一つのネット N のグローバル配線経路(a) とそれに対する信号ピンと接続要求の例(b)

同図 3.2(b)において、信号ピン  $P_2$  および  $P_6$  はゲートの入力端子に対応して設けられたものであるから、各セルの論理的に等価な端子が存在する垂直トラック 8 または 9 のいずれに割り当てても良く、 $P_3$  や  $P_4$  も各貫通セル内の垂直トラック 5 又は 6 のいずれに割り当てても

良い。それに対して、信号ピン  $P_1$  は、ゲートの出力端子に対応して設けられたもので、トラック 2 にのみ置く。このような各信号ピンを置くことができる対直トラックを  $P$  の割当て可能トラックと言ひ、 $P$  の割当て可能トラックの集合を  $T(P)$  と表わす。 $|T(P)| = 1$  であるような信号ピン  $P$  を固定信号ピン、それ以外を浮動信号ピンと言ひ。又、信号ピン  $P$  が属するネットを  $N(P)$  で表わす。

このように、ネットをセルのどの端子に接続すべきか、あるいはネットの配線を貫通セル内のどの垂直トラックに通すべきかという問題を、対応する信号ピンをどの垂直トラックに置くかという問題として表現すれば、本文で考察する端子割当て問題は、各セル行内にあるすべての信号ピンを以下にして重複なく割当て可能トラックに割り当てるかという問題となる。この端子割当てにより、チップ全体の配線問題は、全ての水平チャンネルと左右両端の垂直チャンネルにおける 2 層配線問題に帰着することができる。

ところで、各水平チャンネルでの配線を、一つの層に置かれた水平線分(幹線)およびもう一方の層に置かれた垂直線分(支線)とそれらを結ぶスルーホールによって実現する場合には、上下制約グラフ  $G_v$  で表現される各幹線間の上下制約が生じる。良く知られているように、この上下制約グラフ  $G_v$  がサイクルを含む場合には、配線を実現するために図 3.3(a) のネット 1、3、および 8 のような幹線分割を行わなければならない。このような幹線分割は、スルーホールの個数を増加させ LSI 設計上好ましくないばかりでなく、これらを効率的に扱う良いアルゴリズムが無いから、100%チャンネル配線ができないことが多い。

例えば、図 3.3(a)の端子割り当てを変更し、同図(b)のようにすれば、幹線分割は不必要となり、必要な水平トラック数も減少する。又、チャンネル配線において、必要な水平トラック数はそのチャンネルの配線密度の最大値以上であるから、これを減少させる端子割り当てを行うことも重要である。例えば、図 3.3(a)の左から第 3 番目の端子列上の配線密度は 3 であるが、同図(b)のような端子割り当てを行うことにより、配線密度を 2 とすることができる。

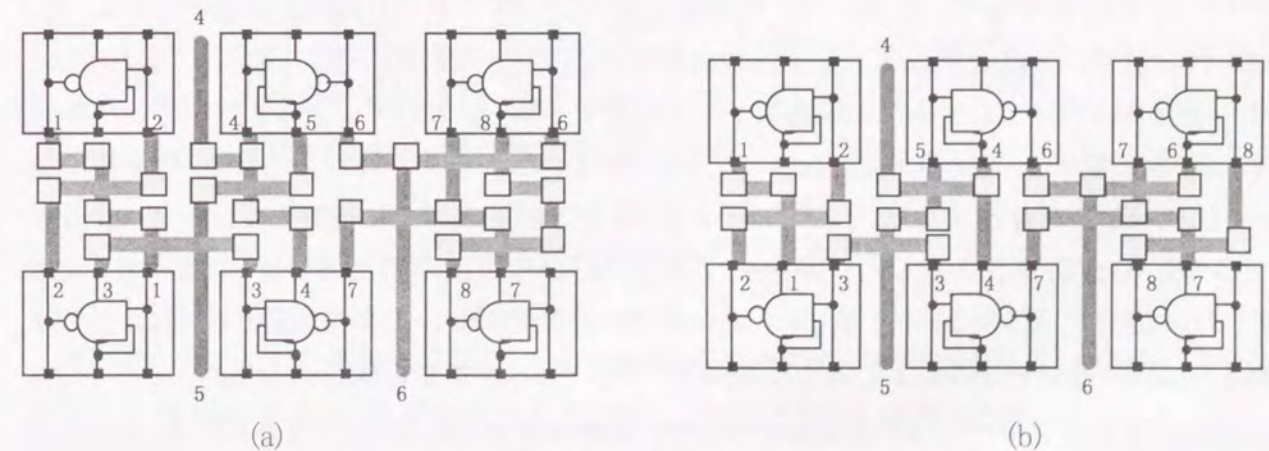


図 3.3 等価端子割当てと必要なトラック本数

以上の点を考慮し、本文では各水平チャンネルにおいて、次の条件を満たす結線要求を生成するような端子割り当てを見出す発見的な手法を提案する。

- (I) 上下制約グラフのサイクルが少ない。
- (II) 最大密度が(チャンネルの容量 -  $\alpha$ )
- (III) 上下制約グラフの辺の個数が少ない。

ここで、チャンネルの容量とは、その水平チャンネル内で利用しうる水平トラックの個数であり、又  $\alpha$  は外部から入力するパラメータで、 $0 \leq \alpha \leq 2$  である。以下では、求める割り当てが条件(I)を満たすことを最優先とし、次に条件(II)を優先する。

### 3.3 等価端子割り当てのアルゴリズム

ここで提案する等価端子割り当て手法は、各浮動信号ピン  $P$  の割り当て可能トラック集合  $T(P)$  の要素を順次減らしてゆき、全ての信号ピンを固定信号ピンとして終了する。

いかでは、まず条件(I)および条件(II)を満たす割り当てを求める手続きについて述べ、その後全体のアルゴリズムを述べる。

条件(I)を満たす割り当てを見出すため、各セル行  $i$  に対して、以下に示すようなラベル付き有向グラフ  $G_L(i)$  を導入する。 $G_L(i)$  の各頂点は、セル行  $i$  に存在する信号ピン  $P$  に対応し、各有向辺  $e = (P_j, P_k)$  とそのラベルを次のように定めるものとする。

(1)  $P_k$  が固定(浮動)信号ピンであり、セル行  $i$  のすぐ上の第  $i+1$  セル行内に、ネット  $N(P_j)$  にぞくする浮動(固定)信号ピン  $P_h$  で  $T(P_h) \supset T(P_k)$  ( $T(P_h) \subset T(P_k)$ ) であるものが存在する場合には、辺  $e = (P_j, P_k)$  を  $G_L(i)$  に付加し、これに正のラベル  $P_h$  を付ける。

(2)  $P_k$  が固定信号ピンであり、第  $i+1$  セル行内に、ネット  $N(P_j)$  に属す固定信号ピン  $P_h$  で、 $T(P_h) = T(P_k)$  であるものが存在する場合には、辺  $e = (P_j, P_k)$  を  $G_L(i)$  に付加し、これに負のラベル  $-P_h$  を付ける。

グラフ  $G_L(i)$  において、正のラベル  $P_h$  をもつ辺  $e = (P_j, P_k)$  が存在するという事は、第  $i$  セル行の信号ピン  $P_k$  と第  $i+1$  セル行の信号ピン  $P_k$  が同一垂直トラックに割り当てられるならば、第  $i$  および  $i+1$  セル行の間の第  $i+1$  水平チャンネルにおいて、ネット  $N(P_j)$  とネット  $N(P_k)$  の間に上下制約が生じることを示している。一方、負のラベル  $-P_h$  を持つ辺  $e = (P_j, P_k)$  が存在するという事は、そのような上下制約がすでに確定してしまったことを示している。従って、グラフ  $G_L(i)$  がサイクルを含み、かつ現在の固定信号ピンの割り当てを変更しないならば、第  $i$  あるいは、第  $j+1$  セル行内の浮動信号ピンの今後の割り当てによっては、第  $i+1$  水平  $rp$  チャンネルにおける上下制約グラフにサイクルが生じることになる。

しかし、 $G_L(i)$  のサイクルが正のラベル  $P_h$  を持つ辺を含むならば、信号ピン  $P_h$  または  $P_k$  のいずれか一方は、浮動信号ピン、他方は、固定信号ピンであるが、その浮動信号ピンの割り当て可能トラック集合から、もう一方の固定信号ピンが割り当てられたトラックを取り除くことによ

り、この辺  $e$  を除去し、上下制約グラフ上にサイクルを生じないようにすることができる。

たとえば、図 3.4(a) に示すような状況に対して、 $G_L(i)$  は、同図(b)のようになり、このグラフのサイクルを解消するために、 $T(P_2) = \{t_2, t_3\}$  から、 $T(P_3) = \{t_2\}$  を取り除くと同図(c)のような信号ピンの割り当てが得られ、ここに対応する  $G_L(i)$  は同図(d)のようになる。

このグラフ  $G_L(i)$  は、手続きの進行に伴って、セル行  $i$  のある浮動信号ピンが固定信号ピンに変化するごとに  $G_L(i)$  を更新しつつ作成する。この更新時に、正のラベルを持つ辺を含むサイクルが生成されたならば、このサイクルを解消するための操作を行い、さらに  $G_L(i)$  を更新する。また、同一セル行の同一トラックに信号ピンが重複して割り当てられることの無いように、固定信号ピン  $P$  が生成されるごとに、 $P$  と同じセル行にある信号  $P'$  で、 $T(P') \supset T(P)$  となるもの全てに対して、 $T(P')$  から  $T(P)$  を取り除くという操作を行う。

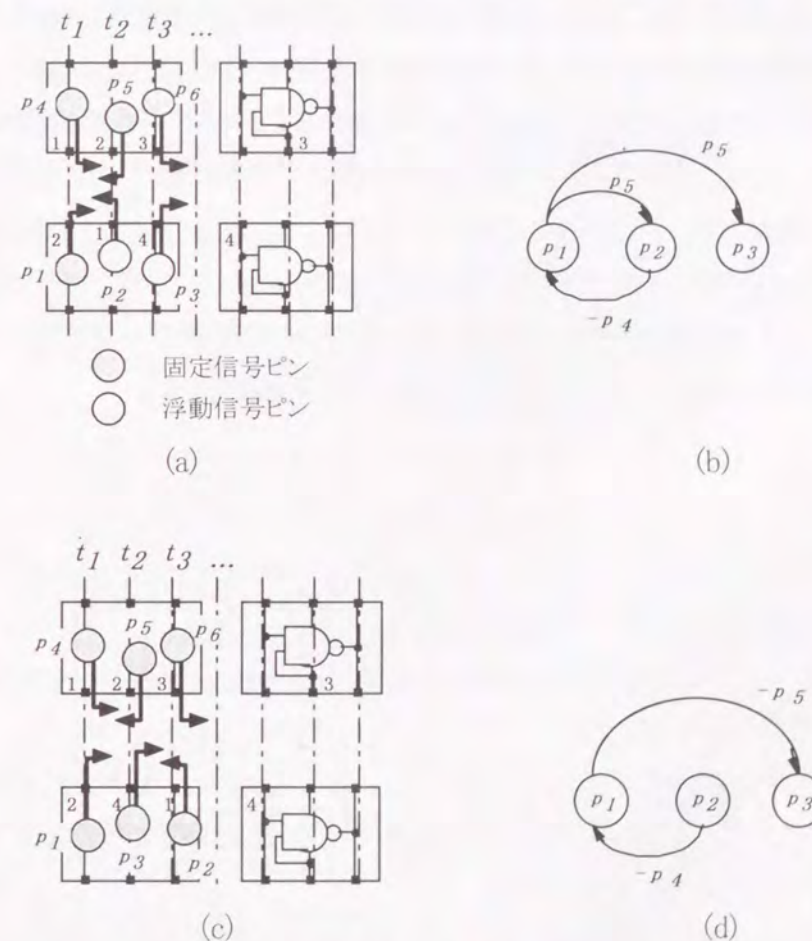


図 3.4 グラフ  $G_L(i)$  に対する処理例

以下に、 $G_L(i)$  を更新する際の一連の操作(手続きFIX)を記述するが、FQは、グラフ  $G_L(i)$  の更新の手続きがまだ実行されていない固定信号ピンを保持するキューである。また、 $SP(i, t)$  はセル行  $i$  の垂直トラック  $t$  上に割り当てることができる信号ピンの集合を示し、 $rp$  は、信号ピン  $P$  が属するセル行を表わす。さらに、 $S(P)$  は、信号ピン  $P$  が属するセル行あるいは垂直チャンネルとしたとき、ネット  $N(P)$  に属しかつセル行  $rp-1$  に含まれる信号ピンの内、

COLにあるもの、COLの左にありCOLにもっとも近いセル列あるいは垂直チャンネル上にあるもの、およびCOLの右にあり、COLに最も近いセル行あるいは垂直チャンネル上にあるものの集合である。図3.2の例では、 $S(P_4) = \{P_3\}$ 、 $S(P_3) = \{P_1, P_2\}$ となり、たとえ $P_2$ の属するセルの右隣のセルにこのネットの信号ピンがあったとしても、それは $S(P_3)$ に含まれない。

<手続き FIX>

1. キュー FQ が空でない限り、FQ から固定信号ピン  $P$  を一つ取り出し、以下の 2. ~5. の操作をおこなう。ただし、 $P$  の割り当てられたトラックを  $t_p$  とする。
2.  $SP(rp, t_p)$  の要素が唯一つであり、かつそれが  $P$  と同ネットに属さない時、それを  $l$  とし、後述の手続き ADJ1 ( $l, P$ ) を行う。それ以外の時、 $SP(rp - 1, t_p)$  の各要素  $l$  について、 $N(l) \neq N(P)$  の時に限り、後述の手続き ADJ2 ( $l, P$ ) を行う。
3.  $SP(rp, t_p)$  の要素が唯一つであり、かつそれが  $P$  と同ネットに属さないとき、それを  $u$  とし、手続き ADJ ( $P, u$ ) を行う。それ以外の時、 $SP(rp + 1, t_p)$  の各要素  $u$  について  $N(u) \neq N(P)$  の時に限り、後述の手続き ADJ2 ( $P, u$ ) を行う。
4.  $SP(rp, t_p)$  の要素の内、 $P$  以外の各要素  $q$  について、 $T(q)$  から  $t_p$  を除去する。このとき、 $q$  が新たに固定信号ピンとなった場合には、 $q$  を FQ に入れる。そうでない場合には、 $SP(rp + 1, t_p)$  の要素が唯一つである時に限り、その要素を  $u$  とし頂点  $q$  に入る  $G_L(rp)$  の辺でラベルが  $u$  のものを全て  $G_L(rp)$  から開放除去する。
5. 1. へ戻る。

<手続き ADJ1 ( $l, u$ )>

信号ピン  $l$  及び  $u$  は共に固定信号ピンであり、 $r_u = r_l + 1$  である。

1.  $G_L(r_l)$  上で  $L$  に入る辺を全て開放除去し、さらに  $S(u)$  の各要素  $a$  に対して、辺  $e = (a, l)$  を作り、そのラベルを  $-u$  とする。
2.  $G_L(r_l)$  上で  $l$  を含む強連結成分[31]  $G_l = [V_l, E_l]$  を求め、 $|V_l| = 1$  であるか、あるいは  $E_l$  中に正のラベルを持つ辺がなければ終了する。あればそのような辺  $e = (a, b)$  を選び、そのラベルを  $x$  とし 3. を行う。
3. 信号ピン  $b$  及び  $x$  の内、浮動信号ピンである方の割り当て可能トラックの集合から、固定信号ピンであるほうが割り当てられたトラックを取り除き、更に、 $G_L(r_l)$  上で  $b$  に入る辺の内、ラベルが  $x$  であるもの全てを開放除去する。この時、新たな固定信号ピンが生成されたならば、それを FQ に入れて 2. に戻る。

<手続き ADJ2 ( $l, u$ )>

信号ピン  $l$  及び  $u$  の内、どちらか一方は、固定信号ピンであり、他方は浮動信号ピンである。

1.  $S(u)$  の要素で  $G_L(r_l)$  において、 $l$  の子孫であるものがなければ、 $S(u)$  の各要素  $a$  に対して、辺  $e = (a, l)$  を作り、そのラベルを  $u$  として終了する。
2. さもなくば、信号ピン  $l$  及び  $u$  の内、浮動信号ピンである方の割り当て可能トラック集合から、固定信号ピンであるほうが割り当てられたトラックを取り除く。この時、新たな固定信号ピンが生成されたならば、それを FQ に入れて終了する。

次に、条件(II)を満たす割り当てを見出す手続きについて考察する。まず、各チャンネルを図3.5に示すような領域に分割し、各領域において、その上下のセル行に属する信号ピンの割り当ての中で、その領域内の最大配線密度を(チャンネルの容量  $- \alpha$ )より大とするような割り当てがある場合にその領域を密領域、そうでない場合にその領域を疎領域とよぶ。

端子割当ての進行に連れて、各信号ピン  $P$  の割り当て可能トラックの集合  $T(P)$  が変化するために、いくつかの領域において、その領域内の可能な最大配線密度が減少し、初め密領域であったものも疎領域に変わることがある。端子割当てが条件(II)を満たすためには、すべての領域が疎領域となって端子割当てが終了しなければならない。

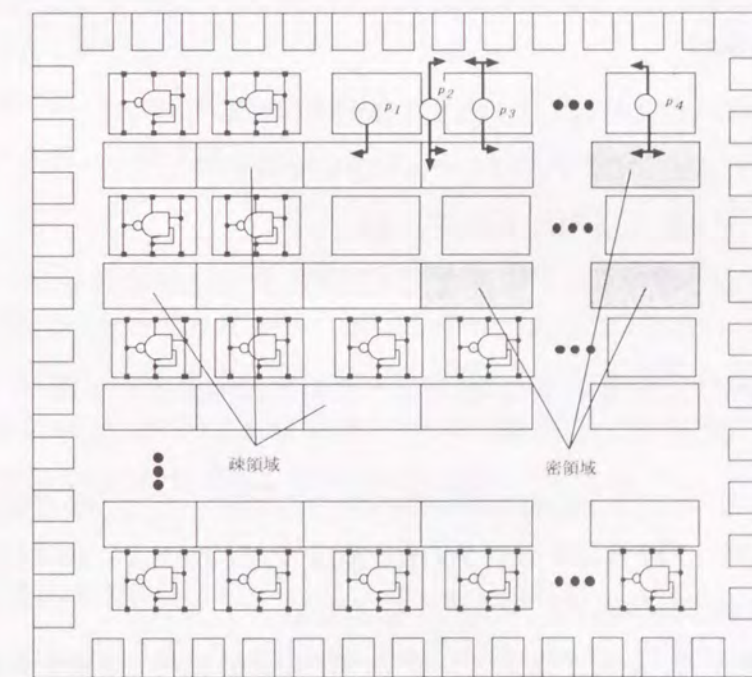


図 3.5 水平チャンネル上の領域

密領域を疎領域に変えるように  $T(P)$  を変更して行くことができるように、各信号ピン  $P$  に対して、 $TYPE$  を次のように定義する。 $TYPE$  は、 $S$ 、 $L$ 、及び  $R$  の3つのフラグからなる集合  $\{S, L, R\}$  の部分集合で、信号ピン  $P$  が存在するセルあるいは貫通セルのすぐ上あるいは、すぐ下の領域が密領域であり、

(i) その密領域で、 $P$  の接続要求が左(右)へ向かう矢印を持ち、かつ、右(左)へ向かう矢印を持たない時、 $TYPE(P)$ はフラグ  $L(R)$  を含み、

(ii) その密領域で、 $P$ の接続要求が左あるいは右へ向かう矢印のみからなる場合には、 $TYPE(P)$ は、フラグ  $S$  を含む。

例えば、図5に示された信号ピン  $P_1, P_2, P_3$ , および  $P_4$  に対して  $TYPE(\cdot)$  はそれぞれ、 $\Phi$   $\{R\}$ ,  $\{S, R\}$ , および、 $\{S, L\}$  となる。

信号ピン  $P$  の  $TYPE(P)$  がフラグ  $R$  を含み  $L$  を含まない場合ならば、 $P$  を  $T(P)$  に属す割当て可能トラックのなかで、できるだけ右に割り当てることにより、 $P$  が存在するセルあるいは貫通セルが接する密領域での配線密度を減少させる可能性のあることがわかる。フラグ  $S$  は、このような割当てを行う際に、どの信号ピンを優先するかを決定するために用いられる。

各信号ピン  $P$  の  $T(P)$  が変化するのとは新しく固定信号ピンができたときのみであら、密領域を疎領域に変更したり、 $TYPE$  を更新したりする操作は、新しく固定信号ピンが生成される毎に行えば良い。その手続きの詳細はここでは、省略するが、手続き **FIX** の操作5のところにも正しく組み入れているものとする。以下に端子割当て手法(手続き **PIN-ASSIGN**)の概略を示す。

#### <手続き PIN-ASSIGN>

[操作1(初期化)] 各セル行  $i$  に対して、以下の操作を繰り返す。

1.  $G_L(i)$  を頂点だけからなるグラフとし、キュー  $FQ$  を空とする。
2. 各垂直トラック  $t$  に対して  $SP(i, t)$  を求める。
3. セル行  $i$  に属する各信号ピン  $P$  に対して、 $T(P)$  及び  $S(P)$  を求め、 $TYPE(P)$  を空とする。

[操作2( $TYPE$ の初期化)] 各領域  $A$  に対して、それが密領域か否かを調べ、密領域ならば以下の操作を行う。

1. 領域  $A$  内で左(右)へ向かう接続領域を持ちかつ右(左)へ向かう接続要求を持たない信号ピン  $P$  に対して、 $TYPE(P)$  にフラグ  $L(R)$  を立て、 $P$  の  $A$  における接続要求が左あるいは右方向のみであるならば、フラグ  $S$  も立てる。

[操作3( $G_L(i)$ の初期化)] 各固定信号ピン  $P$  を  $FQ$  に入れた後、手続き **FIX** を行う。

[操作4(併合の処理)] ある領域において、その上下のセル行に同一ネットに属する信号ピンがあり、それらが共通の割当て可能トラックを持つならば、これらの信号ピンを同一の垂直トラックに割り当てる方が、配線密度を下げ、上下制約を作らないという点で有利である。そこで、このような信号ピン  $P_j$  及び  $P_k$  に対して、常に  $T(P_j) = T(P_k)$  となるような制限を与え、これらが同一の垂直トラックに割り当てられるようにする操作を、 $P_j$  と  $P_k$  の併合という。この併合の操作を全ての密領域に対して行った後、疎領域に対して行う。この時、新たに

固定信号ピン  $P$  が生じたならば、そのたびに  $P$  を  $FQ$  に入れ、手続き **FIX** を行う。

[操作5(浮動信号ピンの位置決定)] 浮動信号ピンが存在していなければ終了。存在するならば、最も多数の密領域に接するセル行を  $i_s$  とし、第  $i_s$  セル行から始めて第  $r$  セル行まで、その後、第  $i_s - 1$  セル行から始めて、第1セル行まで以下の操作を繰り返す。

1. 現在処理中のセル行を  $i$  とし、セル行  $i$  に浮動信号ピンが存在する限り 2. を行う。
2. セル行  $i$  の浮動信号ピン  $P$  で、その  $TYPE(P)$  が  $\{S, L\}$  あるいは、 $\{S, R\}$ ,  $\{L\}$  あるいは  $\{R\}$ 、およびそれ以外のものの集合を、それぞれ  $X, Y$ , および  $Z$  とし、以下の操作を行う。
  - (i)  $X \neq \Phi$  ならば、 $X$  から任意の信号ピン  $P$  を選び、 $X = \Phi$  かつ  $Y \neq \Phi$  ならば、 $Y$  から任意の信号ピンを選ぶ。この  $P$  に対して、 $TYPE(P)$  がフラグ  $L(R)$  を含むならば、 $T(P)$  の中で、最も左(右)のトラックを  $t$  とし、 $T(P) = \{t\}$  とする。
  - (ii)  $X = \Phi$  かつ  $Y \neq \Phi$  ならば、 $Z$  の中から、割当て可能トラックの個数が最小の信号ピン  $P$  を一つ選ぶ。 $T(P)$  の中から、条件(III)をできるだけ満たすようにトラック  $t$  を一つ選び、 $T(P) = \{t\}$  とする。すなわち、 $i \neq i_s$  の場合には、第  $i$  セル行とその直前に 2. の操作が終了したセル行との間の水平チャンネルにおいて、できるだけ上下制約関係が生じないように  $P$  が割り当てられるトラック  $t$  を選ぶ。  
 $i = i_s$  の場合には、任意に  $t$  を選ぶ。
  - (iii)  $P$  を  $FQ$  に入れ、手続き **FIX** を行った後、2. へ戻る。

この手続きのように、信号ピンの割当てを1つずつ発見的に決定していく方法では、処理の途中で  $T(P)$  が空となる信号ピン  $P$  が生じることもある。このような場合には、そのような信号ピン  $P$  が属するセルあるいは、貫通セル内に属する信号ピンを任意に再割当てる。この割当により、このセルあるいは貫通セルの接する水平チャンネルにおいて、上下制約グラフがサイクルを持ったり、併合した方がよいような信号ピンが併合されなかったりする。

#### 3.4 実験と結果の考察

本端子割当て手法を FORTRAN を用いてプログラムし、我々の自動レイアウトシステムに組み入れて、比較的小さな回路に対して実験を行ったところ表1のような結果を得た。図 3.6 は回路  $C$  のチャンネル配線も行った最終のレイアウト結果の一部を示す。使用した計算機は NEC-ACOS1000 である。さらに、本手法の評価を行うために、表1の3つの例題について、乱数を用いて各ピンの割当てを行い、その結果に対してチャンネル配線を施した場合の各水平チャンネルでの配線結果と、本手法を用いた場合の配線結果を比較してみた。その結果を図 3.7 に示すが、ここで、横軸の番号は水平チャンネルの番号であり、縦軸は各水平チャンネルの配線に要した水平トラックの本数とそのチャンネルの容量  $c$  との差 ( $d - c$ ) である。又、丸印  $\bullet$  が乱数による場

合、四角印 ■ が本手法による場合の結果を表わしている。この結果より、本手法が各チャンネルの最大配線密度を減少させるのに有効であることが分かる。特に、同図 3.7(b) の第11および第12水平チャンネルでの結果より、本手法がチップ全体を総合的に考慮している様子が伺える。本手法を用いた場合には、3つの例題のどのチャンネルにおいても100%配線が実現できた。しかし、乱数を用いて割当てを行った場合には、星印 \* をつけた水平チャンネルにおいて、巡回配線要求の原因により100%配線が不可能となったので、各チャンネルから1つのネットを取り除いてチャンネル配線を行った。図7はその結果を示している。また、いくつかのチャンネルにおいて、上下制約グラフの上にサイクルが生じた。そのような水平チャンネルの丸印 ● の傍に、上下制約グラフ中の2個以上の頂点を持つ強連結成分の個数を ( ) に示した。

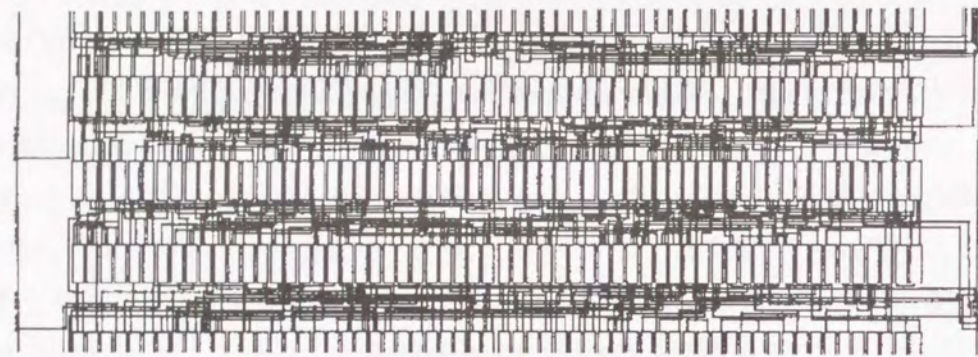


図 3.6 回路Cのレイアウト図の一部

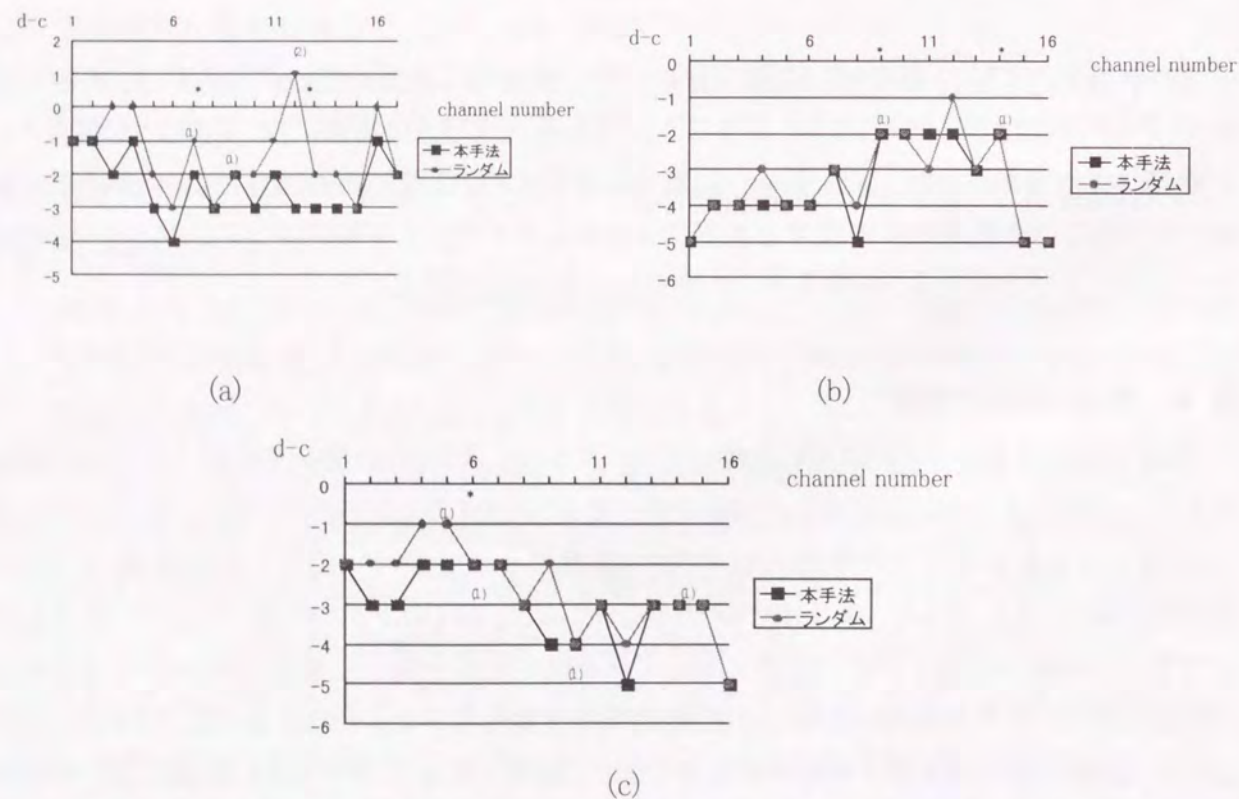


図 3.7 本手法とランダム割当ての場合との比較、(a) 回路 A, (b) 回路 B, (c) 回路 C

### 3.5 結言

本文では、端子割り当ての一手法を提案した。本手法は、各水平チャンネルにおいて、上下制約グラフができるだけサイクルを持たず、かつ、最大配線密度が(チャンネルの容量 -  $\alpha$ ) 以下となるように、最終的に各信号ピンの割当可能トラックを唯一つにして、端子割当てを決定する。この時、ラベル付き有向グラフ  $G_L(i)$  や、フラグの集合  $TYPE(\cdot)$  等を用い、チップのチャンネル全体を総合的に考慮している。

本手法の効果を調べるために、乱数を用いて端子割当てを行った場合との比較を行った。その結果より、本手法の有効性がある程度確認できた。

これまでに発表された自動レイアウトシステムでは、単純な初期割当てとその改善という試行錯誤的な技法によって端子割当てを行うものが多く、またその記述も少ない。その中で、比較的詳細に述べられているものに文献[32]があるが、これでもチャンネル毎に試行錯誤的な技法を繰り返しており、本手法のように、目的(I), (II), および, (III) を満たす端子割当てを全チャンネルを総合的に考慮しながら見出すものではない。



## 第4章 階層的配置配線におけるチップフロアプラン手法

### 4.1 緒言

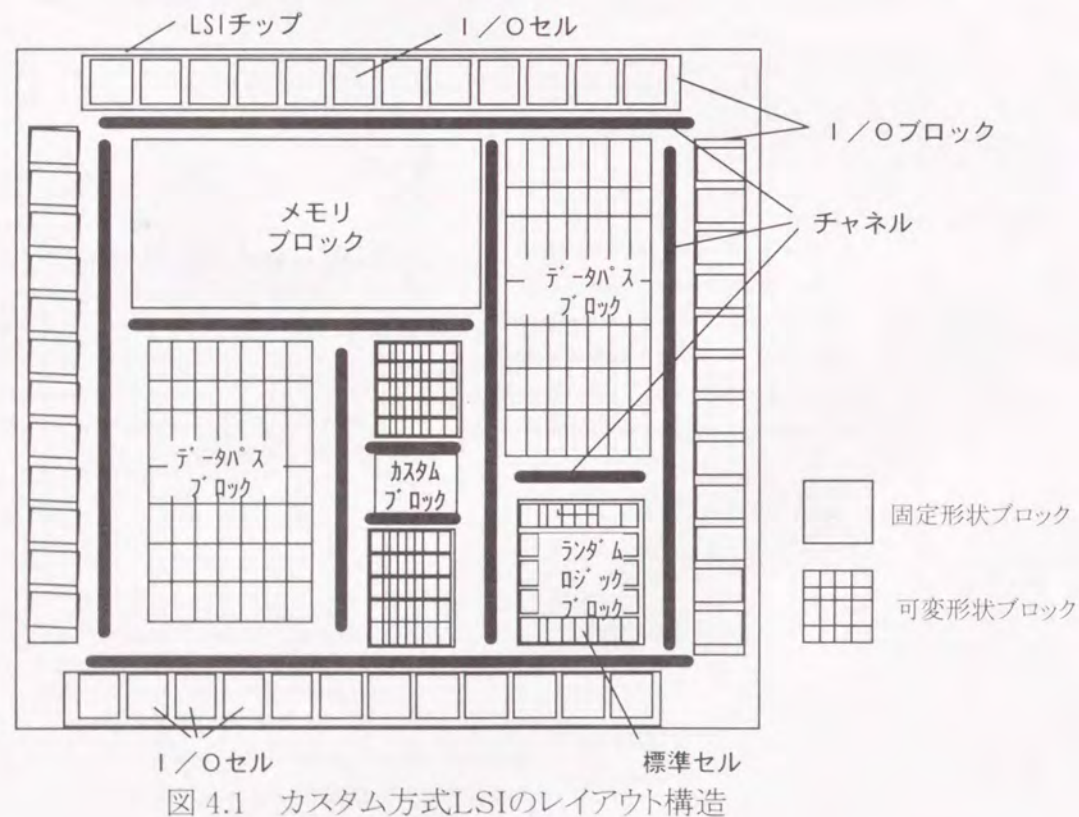
本章では、まず、チップフロアプラン問題の扱うレイアウトモデルについて示す。次に、フロアプランが扱う最適化対象の相互関係について議論し、それに基づきチップフロアプランに適した処理フローを提案する。本論文が提案するフローは、(1)ブロック配置とブロック形状最適化の処理を分割する、(2)ブロック形状最適化では、概略フロアプランの最適化を行った後、詳細フロアプラン最適化を行うものである。

次に、フロアプランの部分問題として、ピン配置・概略配線、電源配線、ブロック形状最適化のそれぞれの手法を説明する。最後に、フロアプラン後チップレイアウトを行ったレイアウト図の一例を示し、実用的かつ良好な結果が得られることを示す。

### 4.2 チップフロアプラン問題

#### 4.2.1 レイアウトモデル

本論文が扱うレイアウトモデルは、図 4.1 に示すものを考える。



チップの周辺には4つの I/O ブロックが存在し、それらに囲まれた矩形の領域に大きさの異なる矩形のブロックが存在する。隣接するブロックあるいはブロック集合に挟まれた帯状の配線

領域をチャンネルと呼び、チップの配線領域全体をチャンネルの組み合わせで表現したものをチャンネル構造と呼ぶ。一般に、ブロックの配置が同じであってもチャンネル構造は一意的ではない。ブロックは、複数の形状を取り得るか否かにより可変形状と固定形状の2種類に分類される。可変形状ブロックは内部のセルや配線、外部ピン位置等が変更可能であるブロックのことであり、標準セルで構成されるブロックや、データバスブロックの一部、等はこれに含まれる。一方、メモリ等のカスタム設計ブロックは固定形状ブロックとして扱う。

チップ面積を評価するためにチャンネルポジショングラフ  $G_{px}$ 、および、 $G_{py}$  を用いる。 $G_{px}$  は図 4.2 に示すように各頂点がブロックの縦辺に、各辺がブロックまたはサブチャンネル(2個の隣接するブロックに挟まれた配線領域であり、チャンネルの一部となる)に対応し、各辺は、重みとしてブロック幅又はサブチャンネルの配線に必要な幅をもつ。グラフ  $G_{px}$  上の最長路長はチップレイアウトに必要な x 方向の最小幅を示しており、x 方向のチップサイズの最小化を行うためには、最長路上の各サブチャンネルが最小幅で実現されることが必要かつ十分な条件である。これらのサブチャンネルはチップ面積の最小化という点で重要な意味を持つので、本論文ではこのようなサブチャンネルをクリティカルサブチャンネルと呼んで区別する。同様に y 方向についても  $G_{py}$ 、クリティカルサブチャンネルを定義できる。

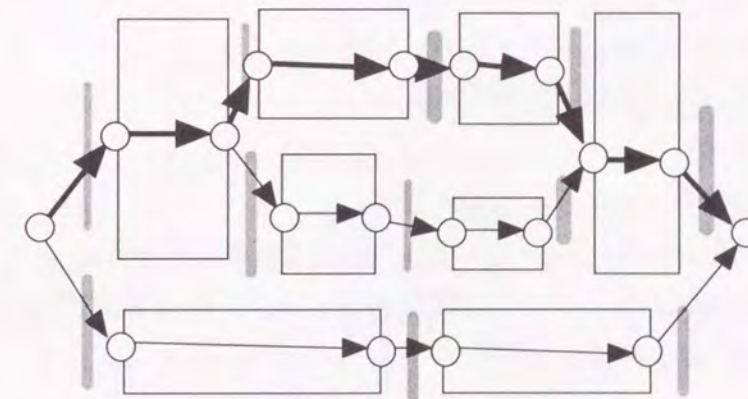


図 4.2 チャンネルポジショングラフ  $G_{px}$

図 4.3 に示すような、ブロック形状の取り得る候補をグラフで表現したものをブロックの形状関数 [34] という。ブロックの高さを縦軸  $H$ 、幅を横軸  $W$  に取り、形状の取り得る高さの値をグラフ上にプロットする。一般のフロアプラン問題において、形状関数による表現は広く用いられている。ところが、形状関数の精度が低ければ、予想してなかった場所にデッドエリア(無効領域)を生じさせることになる。例えば、ブロックの面積・形状は外部ピンの位置などによって最大20%程度変化する。このため、フロアプランの最適化においては、外部ピンの位置などによる影響を考慮した形状関数を高速、かつ、高精度に計算できる推定モデルを準備することが重要である。

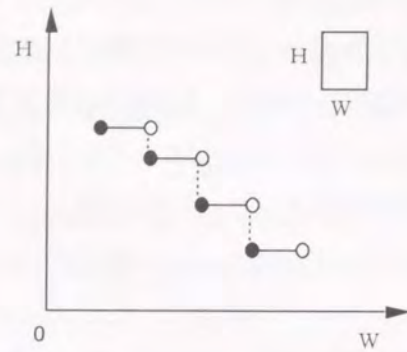


図 4.3 ブロックの形状関数

#### 4.2.2 階層的配置配線フロー

図 4.1 に示したような、階層設計されたLSIを配置配線する場合に、以下に示すような複数の最適化対象と最適化目標を扱う必要が生じる。

##### (i) 最適化対象:

- ・ ブロックの配置
- ・ 可変形状ブロックの外部ピン位置、形状
- ・ 概略配線経路、電源配線経路と幅

##### (ii) 最適化目標と制約

- ・ チップ面積の最小化
- ・ 信号線の遅延制約(各ネットに対して最大遅延制限)の満足
- ・ 電源配線条件制約(ブロックの電源端子における最低電源電位、電源線における最大電流密度の制限)の満足

チップフロアプラン問題は、これらの最適化を大局的に行うことが目的となるが、それぞれの最適化対象は、図 4.4 に示すように、他の最適化対象の影響を強く受ける。この相互依存関係が、チップフロアプラン問題を複雑なものにしている。例えば、ブロック形状を最適化する場合、精度の良いブロック形状関数に基づいて処理を行うことが重要(図 4.4 中 A)であるが、ブロックの面積を精度良く推定するためには、外部ピンの配置が与えられる必要がある。(図 4.4 中 B) しかしながら、ピン配置を最適化するためには、ブロックの形状及びチャネル構造が最終形に近いものが与えられている必要がある(図 4.4 中 C) といったようなサイクリックな相互依存関係がある。

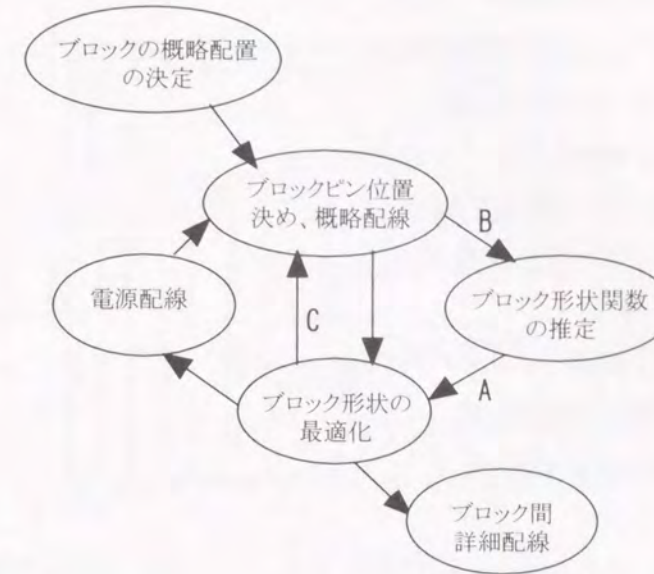


図 4.4 チップフロアプランの最適化対象の相互依存関係

従来のフロアプランシステムでは、ブロックの配置と形状最適化に焦点をあてるものが多く存在した。[35,36] また、ブロック配置を変更しながら、ブロック間配線の推定も変更させるといった技術なども紹介されている。[37,38] しかしながら、これら手法の課題としては、ブロックの配置および形状の決まってない段階での配線領域見積もりを行うため、ブロック形状関数およびブロック間配線の推定精度を上げることが原理的に困難である点である。そのため、フロアプラン処理において、面積最小化等が効果的に行われたとしても、実際にブロックの配置・配線およびブロック間の配線を行ってやると、推定誤差の大きなところで無効領域が発生する。

本論文では、これらのブロック形状関数およびブロック間配線の推定精度を上げることの重要性に着目して、システムの提案と構築を行う。本システム構築時に考慮した主なポイントは以下の2点に集約される。

- (1) 図 4.4 を見ると、ブロック配置、ブロック間詳細配線、それ以外の3つのグループ間にはサイクリックな依存関係がない。また、ブロック形状関数の推定精度は、ブロックの配置および外部ピン配置を前提とすることにより、飛躍的に向上する。このため、最適化問題をブロック配置、形状最適化(狭義のフロアプラン)、ブロック間詳細配線の3つのステップに分割し、順番に処理を行う。
- (2) ブロック形状最適化と概略配線・ピン配置(狭義のフロアプラン)は、最適化対象がサイクリックな相互依存関係になっているため、それぞれの対象を概略の最適化から詳細な最適化へ徐々に移行させ、ブロック形状関数の精度を徐々に上げつつ、概略配線とピン配置も徐々に最適化を進めていく方法を採用。

これらを考慮し、具体的なレイアウト設計フローとしては、以下に示すものを実現する。

- (STEP1) ブロックの概略配置
- (STEP2) 概略ブロック形状関数
- (STEP3) 概略ブロック形状最適化
- (STEP4) 概略ピン配置、グローバル配線
- (STEP5) 高精度ブロック形状関数
- (STEP6) 高精度ブロック形状最適化
- (STEP7) 電源配線
- (STEP8) 詳細ピン配置、グローバル配線
- (STEP9) ブロック内の配置配線
- (STEP10) ブロック間詳細配線

本章の以降の節では、これらの内、STEP2 から STEP8 からなる狭義のフロアプランに関する技術の要素技術として、ピン配置手法および概略配線手法、電源配線手法、ブロック形状最適化手法のそれぞれについて述べる。STEP10 のブロック間配線方法、および、全体のレイアウト合成結果に関しては次章で述べる。STEP1のブロック配置手法に関しては、力学モデル[39]を用いた方法を開発したが、ブロックは多くの場合人手配置が行われる。また、ブロック内の配置配線手法に関しては、本論文では詳細に触れないので、文献を参照されたい。[40,41]

### 4.3 ブロック間配線とピン配置

ピン配置の目的としては、一般的に配線長の最小化が目的とされる[42]が、本論文では、ブロック面積の増大を避けるため、ピン集中の回避も同時に考慮する。ピン配置処理は2段階で行われる。まず、初期配置においては、各々のブロックのピンがブロックの中央に集中していると仮定して、最小経路法によって外部ピンを出す位置を決定する。しかしながら、これでは局所的に端子が集中するため、各々のピンに対して、配置可能な領域と、配置の優先度を定義し、全てのピンの優先度をなるべく満足しつつピンの配置を分散させる方法を用いる。以下に、アルゴリズムを詳しく述べる。

ブロック間の配線領域は、図 4.5 に示すような、チャンネルグラフの各辺を等間隔  $H$  で分割し、それぞれの辺の間に頂点を設定した配線グラフ  $G_r$  を用いて表現する。 $G_r$  の各辺に対応する配線領域を分割領域 (division area) と呼ぶ。ブロックのピンは、まず、ブロック周辺のいずれかの分割領域に割り当てる。未定形状ブロックに対応する面には、中心に頂点を付加し、そこからブロック各辺の中央に最も近い  $G_r$  上の頂点へ仮想枝を付加する。更に、フィードスルーを表現する為に、4边上の頂点が相対する辺同士の頂点を結ぶフィードスルー枝を付加する。

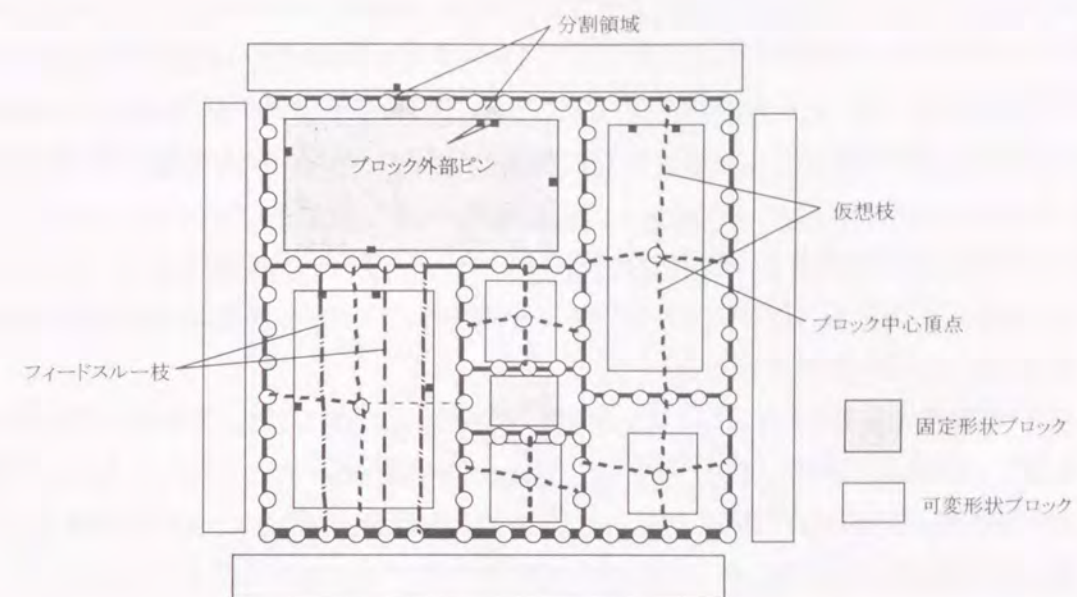


図 4.5 配線グラフ  $G_r$  と分割領域

ピンを分割領域に割り当てるときに、1個の分割領域当たりに割り当てることのできる同一ブロックのピンの個数の上限値を設定する。これをピン配置容量と定義する。また、各分割領域に割り当てた個数をピン配置密度と定義する。

本ピン配置手法は、初期配置、分散配置および詳細配置の3段階より構成される。まず初期配置の段階では、ピン配置容量を無視し、配線長最小化のみを考慮したピン配置を求める。次に分散配置の段階では、ピン配置容量を定義して、ピン配置密度を分散させるための配置改善を行う。最後に詳細配置の段階では、分散領域からピン座標への割り付けを行う。

#### 4.3.1 初期配置アルゴリズム

チャンネル構造が与えられた時に、ピンの初期配置を求める手続き **INIT\_ASSIGN** は、以下の手順で行う。

##### <手続き INIT\_ASSIGN>

1. (配線グラフ  $G_r$  作成) 図 4.5 に示したような配線グラフ  $G_r$  を作成する。フィードスルーを表現する枝は、フィードスルーピンを配置する辺を求める為のものである。同辺には通過可能な本数を示す容量を与える。グラフ  $G_r$  の各辺には、以下のように重みを与える。ブロック間配線領域に対応する辺の重みは、辺に対応した分割領域の長さ  $H$  とする。中心からブロック周辺を結ぶ辺の重みは、(中心から周辺の点までのマンハッタン距離) \*  $a$  とする。 $a$  はピン配置を求める為の探索木が、未定形状ブロックを通過しにくくするためのペナルティを与えるパラメータである。フィードスルーに対応する辺の重みは、(フィードスルーの長さ) \*  $b$  とする。 $b$  はピン配置を求める為の探索木がフィードスルーを通過しにくく

くするためのペナルティを与えるパラメータである。 $a, b$  は共に 1.1 としている。未定形状ブロックのピンの初期位置は、全てのブロック中心の頂点とし、固定形状ブロックのピンの初期配置は、同ピンに最も近い頂点とする。(注:各ピンは最終的に、ブロック周辺上のいずれかの頂に割り当て、フィードスルーネットをフィードスルーを表現する枝に割り当てる。)

2. ネットを囲む矩形の大きさの順で、以下を行う。
  - 1) ネット毎にグラフ  $G_i$  上でのスタイナ木  $T$  を求める。スタイナ木を求めるアルゴリズムは、文献[43] に示された方法を用いる。
  - 2) 以下の処理をスタイナ木  $T$  上で、取り除く枝がなくなるまで行う。 $T$  のリーフとなる頂点  $v$  を探し、同頂点に接続する枝を  $e$ ,  $v$  のもう一方の頂点を  $u$  とする。 $v$  と  $u$  が同じブロック内あるいは周辺の領域に対応した頂点である場合、 $e = (v, u)$  を削除する。
  - 3) ネットに含まれる各ピン  $P_i$  に対して、 $P_i$  を含むブロックの周辺上で、かつ、 $T$  が占める領域をピン  $P_i$  の初期配置とする。

#### 4.3.2 分散配置アルゴリズム

初期配置では、配線長の最小化のみを考慮していた為、ピンが局所的に集中し、ピン配置容量を越える個数のピンが配置された分割領域が点在している。本分散配置は、各分割領域に割り当てられたピンの個数がピン配置容量以下となるようにいくつかのピンを移動し、ピンの配置を分散させることを目的とする。

初期配置からピンを移動する場合には、どのピンをどちらの方向へどれだけの距離を移動すれば良いか、どのピンは初期配置にできるだけ固定したほうが良いか、等を良く考慮し実行することが重要な課題である。本論文では以下のことを考慮し、配置改善を行う。

- (1) クリティカルネットに属するピンは、他のネットのピンより初期配置に優先的に割り当てる。
- (2) ネットの全端子を囲む矩形の周辺上に接するピンは、同矩形の内部に存在するピンに比べて、同ネットの配線長に与える影響が大きい為、それらのピンを割り当てる先としては、同矩形の内部の分割領域を優先する。
- (3) 配線の折れ曲がり数が少なくなるようなピン配置を優先する。
- (4) 1つのネット内で、同一サブチャネル上の分割領域に割り当てられる2端子が存在する場合、これらのピンの座標位置が揃うように配置を調整する。

ピンの分散配置においては、ブロック  $B$  に含まれる各ピンに対して、 $B$  の周辺上の分割領域  $D_1, D_2, \dots, D_n$  のどの領域に割り当てるかという優先度と、各分割領域に割り当てることができるピンの個数を図 4.6 に示すような有向2部グラフ  $G_n(B)$  を用いて表現する。同グラフは各ブロック  $B$  に対して1個を定義する。同グラフの頂点は、配置すべき各ピンに対応した頂点集合  $V_p = \{p_1, p_2, \dots, p_k\}$  と、ブロック  $B$  の各分割領域に対応した頂点集合  $V_d = \{D_1, D_2, \dots, D_n\}$  および仮想頂点  $s$  と  $t$  より構成される。同グラフの各有向枝  $e$  は容量  $c(e)$  を属性としてもつ。頂点  $p_i$  に対応するピンを頂点  $D_j$  に対応した分割領域へ割り当て可能な場合に、

有向枝  $e_{ij} = (P_i, D_j)$  を与え、容量を  $c(e_{ij}) = 1$  とする。各枝には、前述の割当ての優先度をコストへ表現した重みを与える。

ピンの分散配置は、各々のブロック  $B$  のグラフ  $G_n(B)$  に対して、頂点  $s$  から  $t$  への最大フロー[43] を求め、フローが1となる枝に対応したピンの割当てを選択することによって決定することができる。本ピン配置手法によって得られるピン配置とネットの表示例を図 4.7 に示す。これを見れば、ネットの混雑度が解消されていることが伺える。

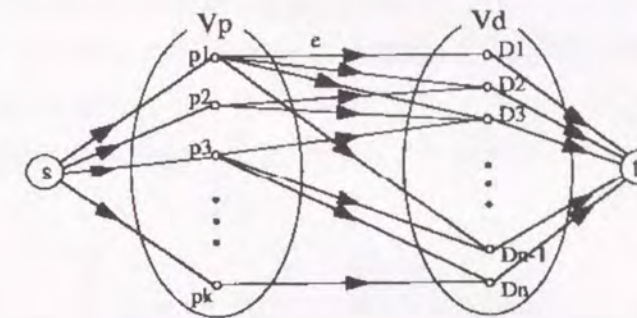


図 4.6 ピン配置を分散させるために用いる2部グラフ

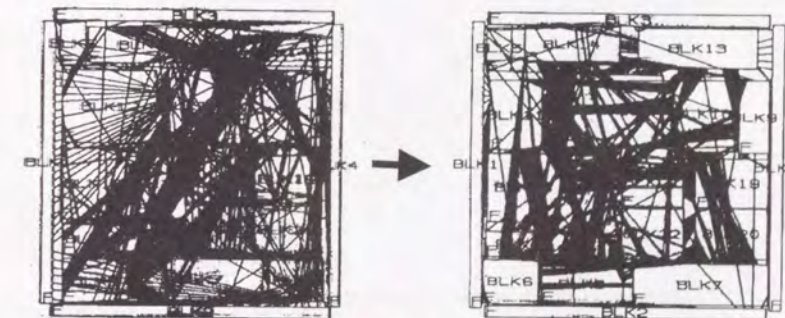


図 4.7 ピン配置による配線混雑度の解消

#### 4.4 電源配線

チップ内の電源配線を行う方法について述べる。電源配線は信号配線に比較して、いくつかの困難な点が存在する。まず、配線の接続が、木構造のみならず一般のグラフ構造を対象とする点である。次に、考えられる全ての動作状態に対して必要以上の配線幅にすると共に、チップ面積削減のために配線幅を最小化する必要がある。

従来提案されている電源配線手法では、木構造のみを扱うもの[44]が、まず提案され、その後、一般的なグラフ構造を扱う数学的手法が提案された。[45,46] しかしながら、動作状態としては、平均動作状態あるいは最大電流のいずれかしか扱うことができなかった。また、配線総面積の最小化を目的としたものであった。

本論文では、以下の点で従来手法を改善した方法について述べる。

- (1) 一般的なグラフ構造を扱う。
- (2) 配線面積の最小化でなくチップサイズの最小化を目的とする。
- (3) 複数の動作状態を考慮して全ての状態でエレクトロマイグレーションや熱断線を引き起こさないよう電源幅を最適化する。

#### 4.4.1 電源配線問題

電源配線の入力としては、図 4.8 に示すような電源配線構造を与える。電源配線構造は、電源パッドからブロックの電源ピンに至る配線の構造として考えられるブロック間全ての枝を与える。これは、冗長な配線要素も含むが、最適化の処理の中で、不要な配線成分は幅を0にして取り除く。更に、表 4.1 に示すように、メモリからのデータ読み出しやデータの変換等の各々の動作状態における各ブロックの消費電力を与える。全ての動作状態の集合を  $BP$  とする。

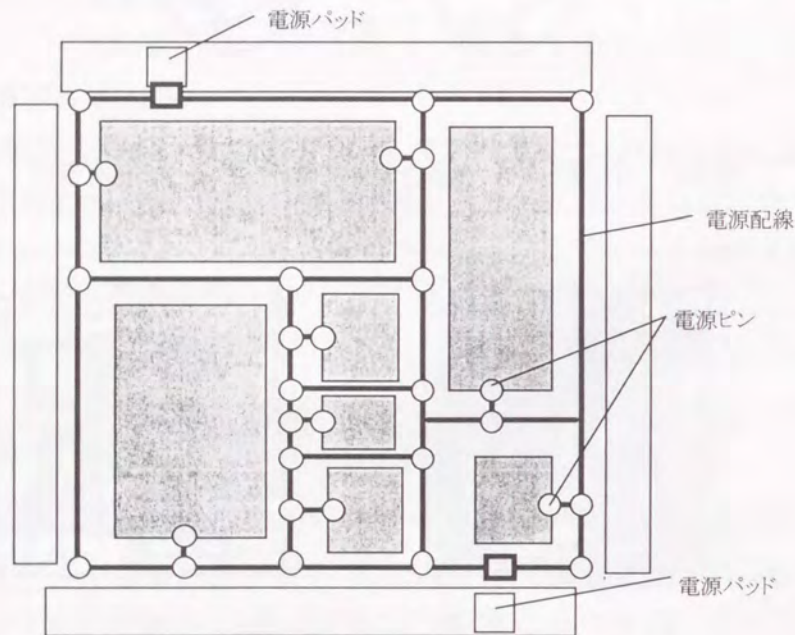


図 4.8 電源配線構造

表 4.1 複数の動作状態におけるブロックの消費電力の例

ブロック	A	B	C	D	E	F	G
最大電流[mA]	50	30	80	28	16	37	55
動作状態1[%]	100	100	0	0	0	0	0
動作状態2[%]	100	100	50	0	30	30	0
動作状態3[%]	0	0	50	100	100	0	100
動作状態4[%]	0	0	0	0	50	50	50

電源配線最適化の目的はチップ面積の最小化である。チップ面積を評価するために図 4.2 に示したチャンネルポジショングラフ  $G_{px}$ , および  $G_{py}$  を用いる。電源配線構造の各枝は、サブチャンネルに対応しており、チャンネルポジショングラフ  $G_{px}$ , および  $G_{py}$  によって、面積に直接影響をあたえるクリティカルサブチャンネルとそれ以外に区別できる。クリティカルサブチャンネルの位置は、電源配線の幅を変更する毎に変更されることがあるが、随時更新が行われているものとする。本電源配線手法においては、クリティカルサブチャンネルに置ける電源配線  $i$  の集合を  $L$  とし、 $L$  に含まれる電源線幅  $w_i$  の総和を最小化することによって面積の最小化を図る。本論文で扱う電源配線問題は以下のように定式化できる。

(i) 目的関数  $F \rightarrow \min.$

$$F = \sum_{i \in L} w_i \quad \text{---(4.1)}$$

$$= \sum (\rho * I_i * I_{ip} / dV_{ip}) \quad \text{---(4.2)}$$

ここで、 $p$  は一つの動作状態であり、 $\rho$  はシート抵抗値、 $I_i$  は電源配線  $i$  の長さ、 $I_{ip}$ 、 $dV_{ip}$  はそれぞれ、電源配線  $i$  に動作状態  $p$  において流れる電流値と電圧降下である。

ただし、エレクトロマイグレーションあるいは熱破壊を起こさない様に電流密度の最大値を制限する。また、電源から各々のブロックの電源端子に至るまでの電圧降下に対しても制限を与える。

(ii) 制約式

電圧降下の制約条件は、電源ピン  $j$  の電位を  $V_{jp}$  としたとき、その上限下限それぞれを  $V_{min}$ 、 $V_{max}$  とすると、次式で表現できる。

$$V_{min} < V_{jp} < V_{max} \quad \text{---(4.3)}$$

また、電流密度の制限は、単位幅あたりの電流密度の上限を  $K$  とすると、次式で表現できる。

$$I_i / w_i < K \quad \text{---(4.4)}$$

すなわち、

$$dV_{ip} < K * \rho * I_i \quad \text{---(4.5)}$$

#### 4.4.2 電源配線アルゴリズム

以下に、電源配線のアルゴリズムを示す。

<電源配線アルゴリズム>

- (1) 十分な幅の電源配線を各辺に与える。

- (2) 目的関数  $F$  の値が変化している間、以下の(3)～(6)を行う。  
 (3) 水平方向と垂直方向のクリティカルサブチャンネルを見つける。  
 (4) 各頂点における電位と各辺における電流を求める。(ここでは、配線幅が十分に取ってあるので、制約条件の式 4.3、式 4.5 は満足している。)  
 (5) 電位を最適化し、電流の分布を変化させずに目的関数  $F$  を最小化する。すなわち、次式 4.6 を、式 4.3 と式 4.7 の制約のもとで最小化する。

$$F' = \sum_{i \in L} \max_p (C_{ip} / dV_{ip}) \quad \text{---(4.6)}$$

ここで、 $C_{ip} = \rho * I_i * I_{ip}$  は定数であり、 $L$  はクリティカルサブチャンネルの集合である。

$$dV_{ip} < K * \rho * I_i \quad \text{---(4.7)}$$

ここで、 $dV_{ip}$  は動作フェーズ  $p$  の時の辺  $i$  の電圧降下である。

- (6) 電位を変化させずに、電流の分布を最適化し、目的関数  $F$  を最小化する。すなわち、次式 4.8 を Kirchoff の電流則(式 4.9)のもとで最小化する。

$$F'' = \sum_{i \in L} \max_p (D_{ip} * I_{ip}) \quad \text{---(4.8)}$$

ここで、 $D_{ip} = r * I_i / dV_{ip}$  は定数である。

$$\sum_{i \in E_j} I_{ip} = 0 \quad \text{---(4.9)}$$

ここで、 $E_j$  は頂点  $j$  に接続した辺の集合であり、 $I_{ip}$  は動作フェーズ  $p$  の時の辺  $i$  を流れる電流である。

ステップ(5)において、制約条件は線形であるが、目的関数は非線形である。この問題は、内部ペナルティ関数を用いることによって、解くことが可能である。[47] ステップ(6)においては、目的関数と制約条件は共に線形であり、線形計画法で解くことができる。

#### 4.5 ブロック形状最適化

ブロックの形状最適化は、与えられた配置(例を図 4.9 に示す。)に対して考えられる全てのチャンネル構造を列挙し(例を図 4.10 に示す。)、それぞれのチャンネル構造に対して、ブロック形状の最適化を行い、面積が最小となるものを選択し、その後ブロックのレイアウトおよびブロック間配線の処理を行いチップレイアウトを完了する。

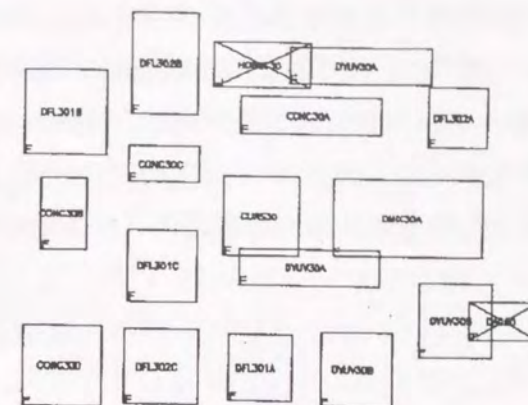


図 4.9 ブロック配置の例

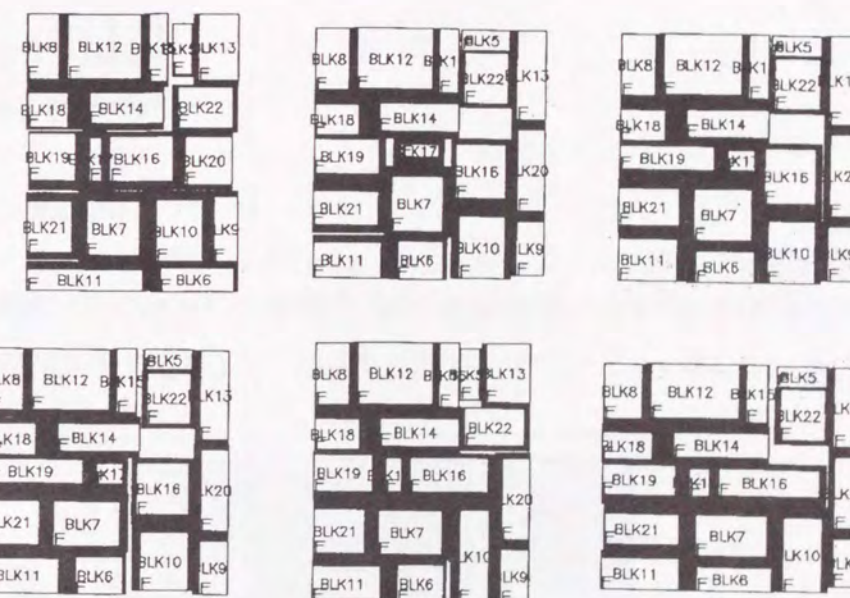


図 4.10 列挙されたチャンネル構造の例

各々のチャンネル構造に対してブロック形状を最適化する方法としては、[48] で提案された方法を使う。概略の手順は、図 4.11 に示す。チャンネル構造として、右下のグラフにある構造を仮定する。この構造の形状関数はこの構造を構成するブロック A, B, C, それぞれの形状関数を組み上げることによって作成できる。まず、水平に隣接したブロック A と B の両方を囲む矩形の形状関数(右上図)は、A, B 各々の形状関数のグラフを水平方向に加え合わせることによって求める。次に、ブロック A と B の両方を囲む矩形とブロック C は、垂直に隣接しているので垂直方向に加え合わせる。このようにして、ブロック A, B, C 全体を囲む矩形の形状関数(右

下図)を求める。次に、右下の形状関数の各ポイントの内、設計目標にあったポイントを選ぶ。設計目標としては、例えば、面積を最小化するポイント、与えられた幅に収まるもので高さを最小化するものなどである。この例では、丸で囲まれたポイントが選択されたとする。このポイントの形状を実現するための、各々のブロックにおける形状は、先程、加え合わせたのと逆方向にポイントをたどって求める。チャンネル構造の列挙は、図 4.12 に示すように、DFS法により、チャンネルを切り領域を2分する操作を階層的に繰り返すことによって求める。[49]

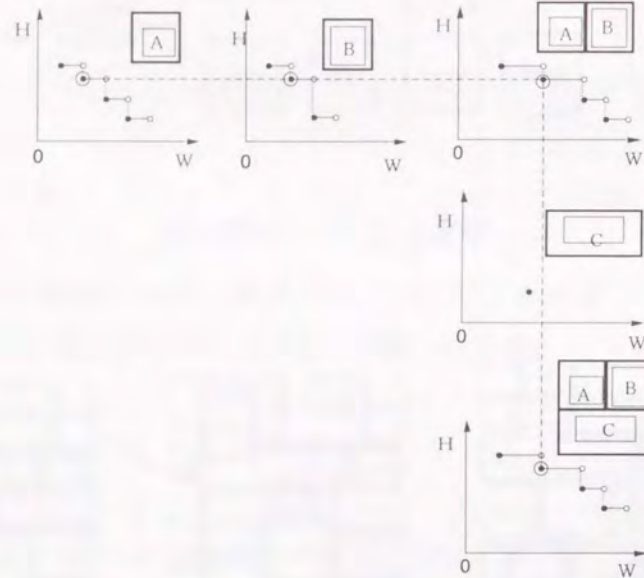


図 4.11 チャンネル構造が与えられた場合のブロック形状最適化

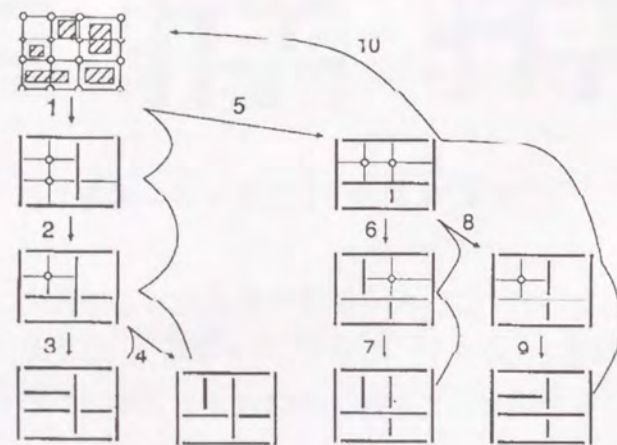


図 4.12 チャンネル構造列挙手法

#### 4.6 ブロック面積の推定 [50]

ブロック形状最適化においては、各々のブロックに対する精度の高い形状関数を求める必要があるが、本論文では、(1)ピン配置が与えられるとする、(2)いくつかの形状に対して、実際にブロック生成を行い、その結果によって形状関数のパラメータフィッティングを行う、といった方法を用いる。従来手法[30-33]では、ブロックの面積を一定と仮定するモデルを用いていたが、実際の標準セルブロックのセル行数を変えて形状を変えた実験によれば、ブロック形状 ( $H$ : 高さ,  $W$ : 幅)において、 $W * H$  の値は一定では無く、セル行数  $R$  の増加に伴い、若干  $W * H$  が、減少する傾向がある。理由としては、ブロック内のローカルな配線は、セルの高さに付着したようなイメージで捉えられ、見かけ上のセル高を高くしたことに相当するが、ブロックに含まれるグローバルな配線による領域は、形状が変化しても、幅方向、高さ方向、共に一定であると考えられる為である。ブロック内のグローバルな配線が占める幅および高さ方向の領域をそれぞれ、 $b, d$  とすれば、ブロックの形状関数は次式で予想できる。

$$W = a / R + b \quad \text{--- (4.10)}$$

$$H = c * R + d \quad \text{--- (4.11)}$$

( $a, b, c, d$  はパラメータ)

#### 4.7 フロアプラン実験例

図 4.9 に本フロアプラン処理を行った後、ブロックおよびブロック間の配置配線を行った結果の例を示す。この図でわかるように、ブロック間の配線領域にあまり無駄な領域が発生していない。配線領域の密度も十分考慮して、形状最適化が行われたことを示している。

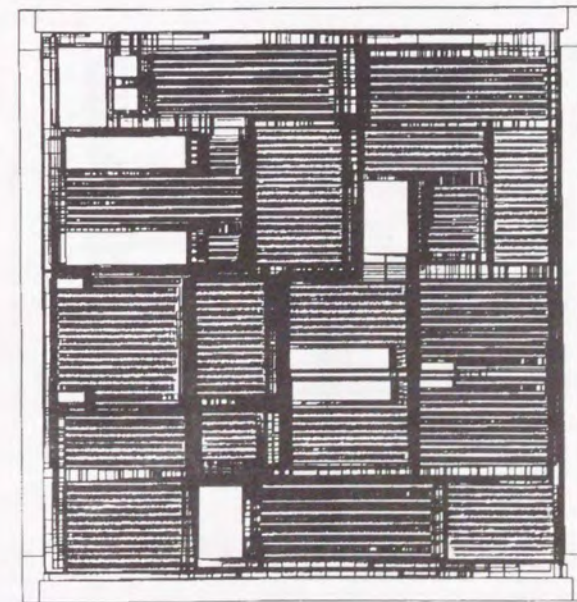


図 4.13 配置配線結果の例

#### 4.8 結言

本章では、チップフロアプラン問題の扱うレイアウトモデルと、フロアプランが扱う最適化対象の相互関係について議論した。チップフロアプランに適した処理フローとして、ブロック配置とブロック形状最適化の処理を分割し、かつ、ブロック形状最適化においては、概略の最適化を行った後、詳細の最適化を行うフローを提唱した。そうすることにより、ブロック形状関数の推定精度を飛躍的に向上させることができ、ブロック間に無駄な領域が無く、各配線チャンネルが効率良く使用した最終レイアウト例を示すことができた。

電源配線、ピン配置、概略配線機能を備え、且つ、これらを考慮した上で最適なチャンネル構造を選択し、ブロック形状を最適化する手法を開発した。ピン配置においては、ピンを分散させる機能により、配線集中を回避した。電源配線に関しては、複数の動作状態における電流条件を満足する電源配線方法を示した。

### 第5章 ブロック間配線手法

#### 5.1 緒言

ブロック間配線を行う方法として、線分探索を用いた手法[6,7]と、チャンネル配線を基本とする方法[8,9]の2種類の構成方法が考えられる。前者はブロック上の空きエリアを自由に通過できるなどのフレキシブルな配線が可能な反面、あらかじめブロック位置を固定してから配線を行うため未配線や、無効領域が生じやすい。逆に後者は基本的に分割統治法であるので、計算複雑度が低く、高速で且つ大規模なデータ処理が行えるといった特徴がある。チャンネル構造が定義できれば、未配線を生じずに配線を完結することができるという大きなメリットがある。

本論文ではチャンネル配線を基本とし、従来チャンネル構造が決定できない場合が存在する問題を解決した。本手法の特徴は、面積最小化とタイミング条件の満足の両方を考慮したグローバル配線、L字型チャンネルを導入しアサイクリックなチャンネル構造を得るチャンネル構成法、および凹凸のあるチャンネルを効率良く配線するチャンネル配線方法等である。

#### 5.2 配線構造の定義

図5.1に示すように隣接するブロックに挟まれた、帯状の配線領域をサブチャンネル、サブチャンネル同士が交差する点をチャンネルクロスといい、サブチャンネルを辺で、サブチャンネルの両端のチャンネルクロスをその辺に接続する頂点で表現したものをチャンネルグラフという。隣接するサブチャンネルを併合して得られる帯状の配線領域をチャンネルといい、チャンネルグラフをチャンネルの集合で行現したものをチャンネル構造という。

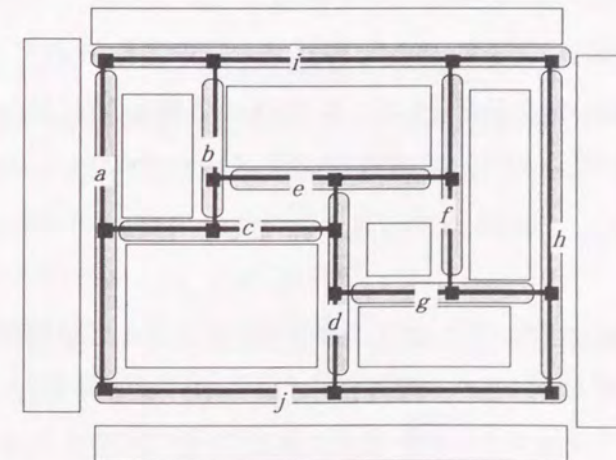


図5.1 チャンネル構造の例

チャンネルが交差する状況に応じてチャンネル配線の処理順に制約が生じる。たとえば、チャンネル *e* とチャンネル *f* は T 字型に接しているため、*e* の方を先に処理しなければならない。このよ



うな処理順をチャンネル処理順といい、この順序制約をグラフで表現したものをチャンネル処理順グラフ  $G_c$  という。図 5.1 に対応する  $G_c$  の例を図 5.2 に示す。

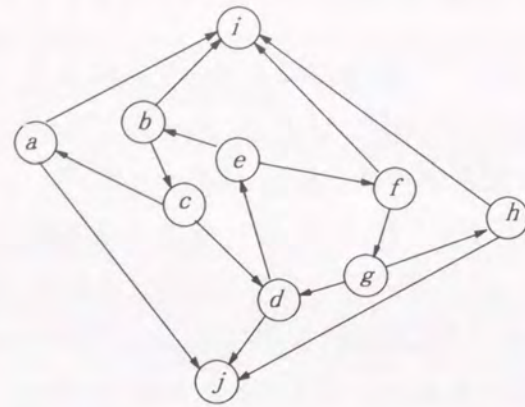


図 5.2 チャンネル処理順グラフ  $G_c$

本手法の配線処理は、各ネットの配線が通過するサブチャンネルの集合を求める概略配線、その後チャンネル構造を作成しチャンネル処理順にしたがって各チャンネルの配線を行う詳細配線の2段階で行う。

### 5.3 概略配線

#### 5.3.1 処理概要

概略配線の処理は以下の手順で行う。

1. (サブチャンネルの構成) チャンネルグラフを作成する。
2. (電源グローバル配線) 電源ネットの配線経路と配線幅を決定する。
3. (初期配線) 面積最小化を目的として、各ネットの概略配線経路を求める。
4. (再配線) 面積最小化および配線遅延値の最適化を目的として再配線を行う。

#### 5.3.2 初期配線

ここでは、ブロック間配線領域を図 5.3 に示すようなグローバル配線グラフ  $G_R$  でモデル化する。 $G_R$  は各サブチャンネルを間隔  $d$  ( $d$  は外部からパラメータで与える) で等間隔となるように分割し、各分割点およびチャンネルクロスを頂点对応させ、隣接する分割点またはチャンネルクロス  $a, b$ , 間の辺  $e = (a, b)$  に対応付けたものである。ブロック上のピンは最も近い分割領域に属しており、各ネットのグローバル配線はこれらの分割領域の集合で表わされる。各分割領域  $a$  の配線に必要な幅  $w(a)$  は、ブロックのどっぴり及び、分割領域のを通過する配線できるだけ詰めたときに必要な幅のことをいう。各クリティカル領域 (critical division area) 上で、 $w(a)$  を最大にする分割領域をクリティカル分割領域という。クリティカル分割領域に新たな配

線が通過するとチップレイアウトに必要なサイズが増大するので、できるだけこのような配線を避けることが好ましい。そこで、 $G_R$  のクリティカル分割領域に対応する辺のコストを他の辺のコストより大きくなるように与え、最小コストパス法により配線経路を求める。

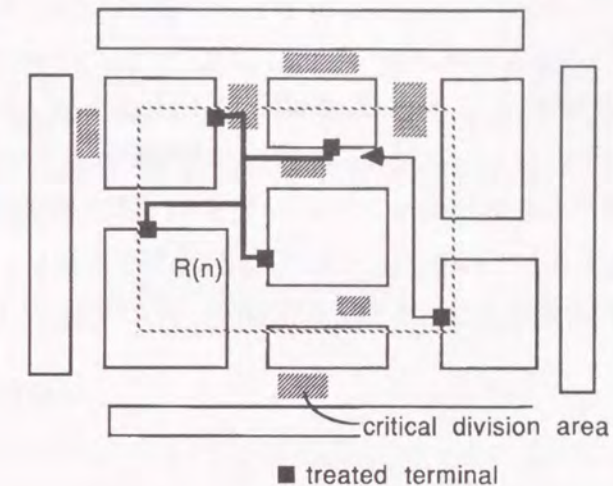


図 5.3 グローバル配線グラフ  $G_R$  とクリティカル領域

#### <初期配線アルゴリズム>

1. (初期化)  $G_R$  を作成し、電源配線された状態を初期状態として各分割領域の配線密度及びクリティカル分割領域を求め、クリティカル分割領域に対応した辺にコスト  $C + \alpha$  ( $C \gg d$ ) を、他の辺には  $d$  を与える。
2. (ネットの順序づけ) 全ネットを、ネットの全端子を含む最小矩形の周辺長の昇順でソートする。
3. (経路探索) 2. の順で各ネット  $n$  の配線経路を求める。
  - 3-1. ネット  $n$  の全端子を囲む最小矩形の中心に最も近い端子を  $p_0$ , 既配線端子集合  $P$  を  $\{p_0\}$ , グローバル配線  $R(n)$  を空とする。
  - 3-2. 集合  $P$  の全端子を囲む最小矩形の最も近い未配線端子  $p_i$  から  $R(n)$  への最小コストパスを求めそれを  $R(n)$  に付加する。
  - 3-3. この操作を未配線端子がなくなるまで繰り返す。
  - 3-4. 各分割領域の配線密度、クリティカル分割領域、コストを更新する。

#### 5.3.3 再配線

初期配線のようにネットを順次配線する方法では、クリティカル分割領域も順次変更されるため、初期に処理したネットの配線は、配線後のクリティカル分割領域を十分考慮していない。そこで、クリティカルなサブチャンネルに属し、かつ、2個以上にサブチャンネルにわたるネットの配線を引き剥がし、初期配線段階3の処理で再配線を行う。(手続き 再配線 A)

次に、全ネットに対して、出力ピンから入力ピンに至るパス毎に信号遅延時間を計算し、あらかじめ与えられた要求値(最大許容遅延時間)の70%を超えるパスを含むネット  $n$  に対して以下の再配線手続きを行う。

#### <手続き 再配線 $B(n)$ >

ネット  $n$  のグローバル配線を  $R_1(n)$ ,  $G_R$  の辺のコストを全て  $d$  として初期配線のステップ 3. の処理で求めた配線を  $R_2(n)$ , 同じ  $G_R$  に対して初期配線ステップ 3. で出力端子を  $p_0$  とし、遅延の大きい入力端子から順に最小コストパスを求めて得た配線を  $R_3(n)$  とする。 $R_1(n)$ ,  $R_2(n)$ , ...,  $R_3(n)$  それぞれについて遅延時間を計算し、ネットに含まれるパスの(遅延時間/要求値)の最大値を最小とする配線を選び、元の配線  $R_1(n)$  と置き換える。

### 5.4 詳細配線

#### 5.4.1 処理概要

詳細配線の処理は以下の手順で行う。

1. (ブロックの位置決め)面積最小化を目的として、ブロックの位置決めをおこなう。
2. (チャンネル構成)アサイクリックなチャンネル処理順となるように直線および L 字型チャンネルによるチャンネル構造を求める。
3. (チャンネル配線)チャンネル処理順に従って各チャンネルの配線をおこなう。

#### 5.4.2 ブロックの位置決め

チップ面積を最小化するためには、各クリティカルサブチャンネルを最小幅で実現すること(条件 A)が必要かつ十分であることは既に述べた。クリティカルサブチャンネルを含むチャンネルにおいて、クリティカルでないサブチャンネル部分の配線余裕がなく、しかもその部分の配線がグローバル配線の見積り幅より広い領域を必要とした場合には、クリティカルサブチャンネル部分での空き領域が生じ条件 A を満足しなくなる。このため、x, y, 両方向について、条件 A を満足するためには、クリティカルでないサブチャンネルの配線余裕を分散させる必要がある。

ここでは、ブロック  $B, B'$  間のチャンネル幅、配線に必要な幅を  $d(B, B')$ 、 $w(B, B')$ 、配線余裕度を、

$$a(B, B') = (d(B, B') - w(B, B')) / w(B, B') \quad \text{---(5.1)}$$

で定義し、各ブロック  $B$  について、左側に隣接するブロックで、配線余裕度を最小にするものを  $B_l$ , 右側で同様のものを  $B_r$  としたとき、 $B_l$  および  $B_r$  それぞれとの間の配線余裕度が等しくなるような配置を求めることを目的とする。(y 方向についても同様)

#### <ブロック位置決めアルゴリズム>

##### ブロックの位置決定

各々のブロックの両側における領域の広さに対する配線面積の比を均一化する。

1. 各ブロックが移動しなくなるまで以下をおこなう。
  - 1-1. 各ブロック  $B$  について  $a(B, B_l)$ ,  $a(B, B_r)$  を求める。
  - 1-2. ブロック  $B, B'$  間のチャンネル幅を  $d(B, B')$  とするとき、各ブロック  $B$  を  $\Delta x(B) = \min(d(B, B_l), d(B, B_r)) * (d(B, B_l) - d(B, B_r))$  だけ左へ移動する。
  - 1-3. y 方向についても 1-1, 1-2. と同様の移動を行う。

#### 5.4.3 チャンネル構造決め

チャンネル処理順グラフ  $G_c$  にサイクルがある場合、サイクル上のチャンネルを選び、チャンネルがクロスする部分をスイッチボックスとし、スイッチボックスでチャンネルを分割することによりサイクルを除去する手法が提案されているが、この手法は100%配線を保証できない。このため、本論文では L 字型チャンネルを含むアサイクリックなチャンネル構造を構成し、各チャンネルを順次配線することにより100%配線を保証する手法を用いる。

$G_c$  上にサイクルがある場合、サイクル上の T 字型チャンネルクロスを L 字型および直線チャンネルに分割し、サイクルを除去する(図5)。このとき、L 字型チャンネルは配線が困難なため、できるだけ配線が容易な部分で構成するのが望ましい。本手法ではサイクル上のチャンネルの内、L 字型チャンネルを構成可能な部分をさがし、それが複数個ある場合は、構成される L 字型チャンネルの配線が容易な領域において構成する操作を繰り返すことにより、アサイクリックな  $G_c$  を得る。

##### < $G_c$ のサイクル除去アルゴリズム>

1.  $G_c$  の強連結成分を求める。
2. 頂点数が 2 以上の強連結成分  $G_{st}$  がなくなるまで、以下を行う。
  - 2-1.  $G_{st}$  に含まれる有向枝  $(u, v)$  の内、 $u, v$ , とともに  $G_{st}$  に属する頂点に向かう有向枝の本数が 1 であるものを  $E^*$  に加える。ただし、終点  $v$  を共通とする始点  $u$  が複数個ある場合は、 $v$  の子  $s$  と  $u$  のチャンネル間隔  $d(u, s)$  が最小の物のみを  $E^*$  に加える。
  - 2-2.  $E^*$  の枝の内、次の評価関数が最小となるものを選ぶ。ここで、 $u, v'$  は枝  $e$  に対応した T 字型チャンネルクロス部で形成される L 字型チャンネル  $u-v'$  の直線チャンネル部分である。

$$W(e) = C * \alpha + \beta \quad (C \gg \beta)$$

- $$\begin{aligned} \alpha &= 0: u, v' \text{ ともにクリティカルでない} \\ &= 1: u, v' \text{ のうち1つだけがクリティカルである} \\ &= 2: u, v' \text{ ともにクリティカルである} \end{aligned}$$

$\beta$ :  $u, v'$  のうち短い方のチャンネルの長さ

2-3. チャンネル  $u, v$  を L 字型チャンネル  $u-v'$  と直線チャンネル  $v''$  に分割し、グラフ  $G_c$  を更新する。

本手法の処理例を図 5.4 に示す。

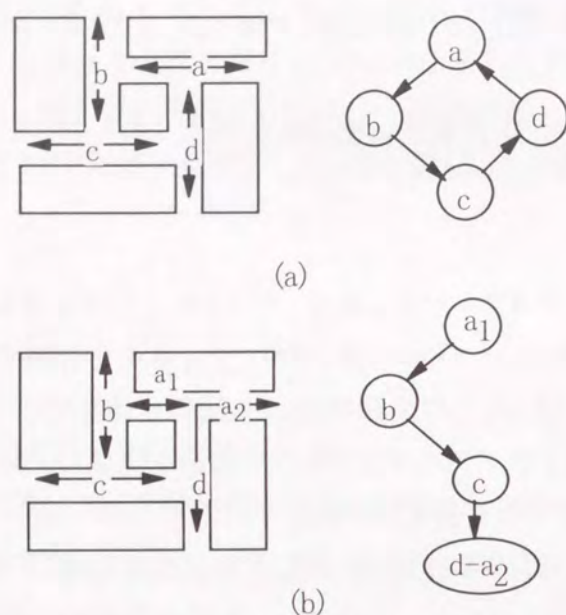


図 5.4 チャンネル構造決めとグラフ  $G_c$  の例

#### 5.4.4 チャンネル配線

従来の多くの配線システムは、配線領域を格子で表現し、配線格子に配線やコンタクトを置く方式がとられている。この方式は簡便である反面、チャンネル配線に必要な幅が増大する欠点がある。そこで、本論文では配線格子の制約を受けないグリッドフリー方式を採用した。加えて、L字チャンネル配線にも対応するため、次の特徴を持たせた。

- (1) 凹凸のあるチャンネル形状の配線が可能。
- (2) 三方が固定されたチャンネルの配線が可能。

<チャンネル配線アルゴリズム>

1. (初期配線トポロジーを求める) 同ネットの隣接する端子間に1本の水平線分(幹線)を配し、幹線の両端(端子点)から端子に向かって垂直線分(支線)を引く。ただし、この段階では、まだ各幹線の位置はまだきまっていない。
2. (重みつき制約グラフ  $G_w$  の作成) 幹線間の位置制約を示すグラフ  $G_w$  を作成する。  $G_w$  の重み  $w$  のついて有向辺  $(a, b)$  は、幹線  $a$  を幹線  $b$  より  $w$  またはそれ以上

上に配置しなければならないことを示し、重み  $w$  のついて無向辺  $(a, b)$  は、幹線  $a, b$  に上下関係の制約はないが、距離  $w$  またはそれ以上離さなければならないことをしめす。さらに、チャンネル形状の各水平線分およびチャンネルの上端  $S$ 、および下端  $T$  に対応した頂点を  $G_w$  に付加し、これらの間および、これらと各幹線との間の位置制約を示す有向枝を  $G_w$  に付加する。

3. (幹線分割)  $G_w$  にサイクルがあるばあい、チャンネル形状に凹凸がある場合、  $G_w$  に長いパスがある場合、固定された幹線がある場合等には、チャンネル配線が不可能になったり、配線に必要な幅が大きくなったりするので、配線トポロジーおよび  $G_w$  の修正を行う。
4. (幹線割り当て) 各幹線を後述の手続き TRUNK\_ASSIGN によって割り当てる。
5. (メタル最大化) アルミ、ポリシリコンの2層配線の場合、支線のポリシリコン部分をできるだけアルミに置き換える。

<手続き TRUNK\_ASSIGN>

- (1)  $G_w$  上で  $S$  から各頂点  $a$  への最長路長を  $u(a)$ 、  $a$  から  $T$  への最長路長を  $l(a)$  とする。
- (2) 未割り当ての頂点の  $u$  値、  $l$  値の最小値をそれぞれ  $umin, lmin$  とする。  $S_u$  を  $umin < u(a) < umin + VCL$  ( $VCL$  は水平線分とコンタクト間の最小矩形) を満足する未割り当て頂点  $A$  の集合、  $S_l$  を  $lmin < l(a) < lmin + VCL$  を満足する未割り当て頂点  $B$  の集合とする。
- (3)  $S_u, S_l$  がともに空でない限り以下を行う。
  - (3.1)  $S_u$  (または  $S_l$ ) 中に  $S_u$  (または  $S_l$ ) 中の他の頂点に向かう無向枝のない頂点あるいは、固定幹線に対応した頂点  $v^*$  をチャンネルの上辺(または下辺)から距離  $u(v^*)$  (または  $l(v^*)$ ) の場所に割り当てる。
  - (3.2)  $S_u$  (または  $S_l$ ) 中で、以下の値  $Q(v)$  を最大にする頂点  $v$  をチャンネルの上辺(または下辺)から距離  $u(v)$  (または  $l(v)$ ) の場所に割り当てる。

$$Q(v) = -C1 * u(v) + C2 * l(v) + C3 * h(v) + C4 * p(v) \quad \text{---(5.1)}$$

( $v$  が  $S_u$  に含まれる場合)

$$Q(v) = -C1 * l(v) + C2 * u(v) + C3 * h(v) - C4 * p(v) \quad \text{---(5.2)}$$

( $v$  が  $S_l$  に含まれる場合)

ただし、  $C1$  から  $C4$  は定数であり、  $C1 \gg C2 \gg C3 \gg C4$  である。

ここで、 $h(v)$  は、チャンネル左端から、頂点  $v$  に対応した幹線の右端までの距離、 $p$  はチャンネル上方への割り当て優先度である。(両端の支線が上方(または下方)に接続しているものに 1 (または -1)、その他場合に 0 を与える)

(3.3)  $G_w, umin, lmin, S_u, S_l$  を更新する。

### 5.5 実験と結果の考察

図 5.5 は、グローバル配線および、ブロック位置決めを行った結果である。各サブチャンネル内の配線に必要な幅を図示し、クリティカルでないサブチャンネルの配線余裕度が分散している様子が伺える。

本ブロック間配線を実際のいくつかの VLSI 回路に適用した結果を表 5.1 に示す。計算機は IBM3081, プログラミング言語は FORTRAN77 を使用した。ここで、グローバル配線での面積は、配線間隔を VCL(コンタクトと配線の間隔)として見積もっている。再配線 A により、多少面積の改善が得られている。グローバル配線と詳細配線結果の面積差がわずかであるのは、詳細配線で最適解に近い解が得られていることを示している。図 5.6 は回路 A について、再配線 B 前後の遅延時間分布の変化を示す。縦軸は要求値と計算結果の比、横軸はネット番号である。いくつかのネットについて遅延時間が改善されており、再配線 B 以前に遅延計算値と要求値との比が最大 111%であったのが、最大 82%にまで減っている。

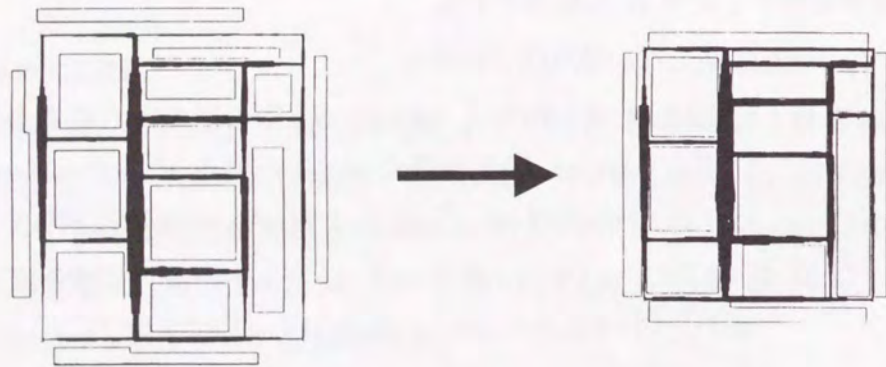


図 5.5 ブロック位置決め例

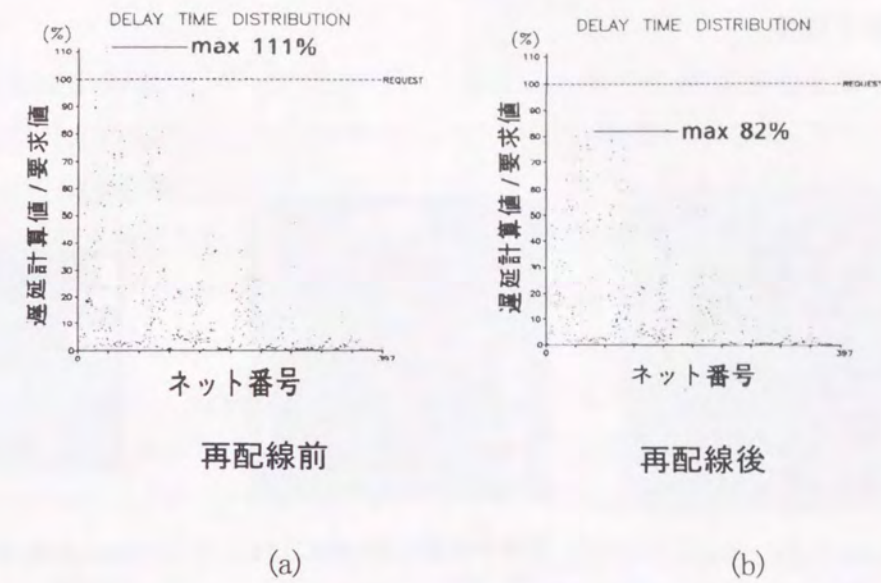


図 5.6 概略配線による遅延条件の改善

表 5.1 ブロック間配線結果の例

回路名	A	B	C	
ブロック数	14	15	11	
ネット数	397	337	414	
ピン数	1434	980	1142	
ゲート数	10000	6200	3800	
面積	概略配線	54.49mm <sup>2</sup>	34.05mm <sup>2</sup>	58.09mm <sup>2</sup>
	詳細配線	54.74mm <sup>2</sup>	34.05mm <sup>2</sup>	58.13mm <sup>2</sup>
計算時間	概略配線	43 sec	20 sec	47 sec
	詳細配線	74 sec	12 sec	62 sec

本レイアウト合成システムは、実際のLSI約30品種の開発において適用された。図 5.9 に、全自動でレイアウト合成を行った LSI の内、代表的な結果を示す。(使用マシン:DEC mini computer 1MIPS 程度)

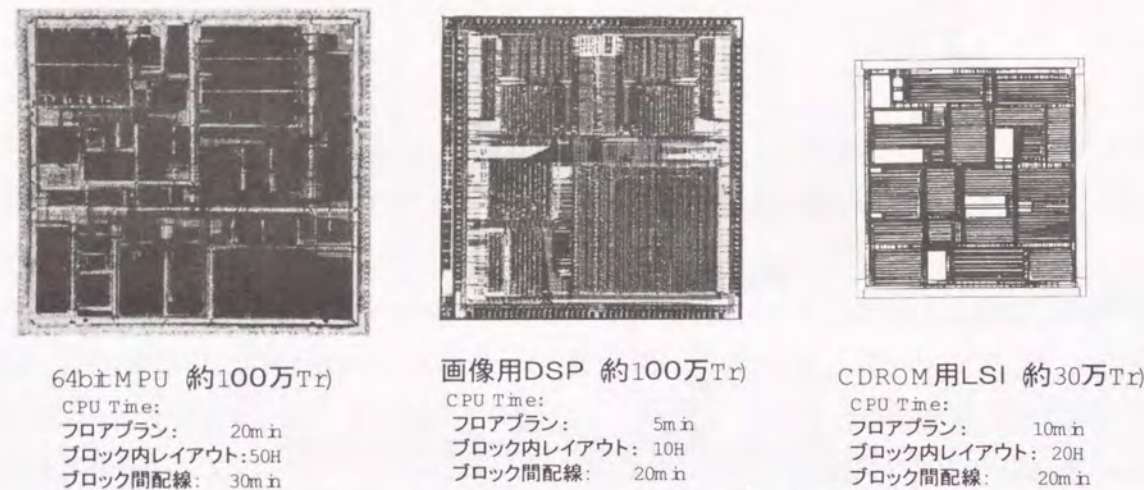


図 5.7 カスタムLSIのレイアウト合成例

## 5.6 結言

本章では、ブロック間配線に対するアプローチとして、初期配線後、面積改善と遅延改善のための再配線を行うことで、面積最小化とタイミング条件の満足の両方を考慮したグローバル配線手法について述べた。実験結果により、面積増を極力減らしながら、遅延条件を大幅改善する結果を得た。

詳細配線は、チャンネル配線を基本とするが、L字型チャンネルを導入しアサイクリックなチャンネル構造を得るチャンネル構造決定法と凹凸のあるチャンネルを効率良く配線するチャンネル配線方法によって、常に100%配線を行える方法を提案した。

実験により短時間で遅延特性の良いレイアウト結果が得られることを確認した。また、実際の大規模集積回路への適用例により実用面でも活用され、良好な結果を得ていることを示した。

## 第6章 標準セルの合成手法

### 6.1 緒言

従来、リーフセルの面積は、チップ面積ひいてはチップコストに直接的な影響を与えることから、人手による設計が行われていた。昨今のライブラリ開発頻度の増加によりライブラリ自体のライフサイクルタイムは短期化の様相を示しているため、設計期間の点からも設コストの点からも大幅な削減が要望されており、人手コストを十分にかけて開発することが困難となりつつある。以上のような理由から、テクノロジーフリーなライブラリセル合成技術の重要性はますます高まりつつある。しかしながら、従来のセルレイアウトの合成方法では、計算機処理を簡単化するために、様々な制約を有し、それらの制約がプロセスに対する独立化や、人手設計セルの流用や、レイアウトの最適化を妨げる場合が多い。例えば、上原、他 [15] を始めとする多くのリーフセル合成システムは、トランジスタを1次元配置したモデルを用いるため、一般に人手設計されたリーフセルの配置情報をコンパクションによって再利用することができない。

そこで、本論文では、[53] において、トランジスタの2次元配置モデルをセル合成に導入し、トランジスタ配置に関しては人手設計並みのフレキシビリティを有するレイアウトモデルに対するセル合成方法を提唱した。今回、更に、トランジスタ上のメタル引き出しコンタクトの位置・形状最適化や、トランジスタ上のメタル配線通過を可能とし、且つ、ゲートの折曲げを取り入れることによって、より柔軟にプロセス変化に対応できるレイアウトモデルと、そのレイアウト合成方法を報告する。

レイアウトデータの抽象化と言う概念に関しては、従来あまり論じられることがなかった。ほとんどの場合、マスクレイアウトの方式はルール [54] あるいは、単に図形間のスペーシングルール等が変化するのみであると仮定されており [55,56,57]、レイアウトの構造変化(例えば、配線層の増加や、トランジスタ拡散領域の製造方法の変化によるレイアウトの自由度増大等)は考慮に入れられてなかったためである。一例、として、シンボリックレイアウトの手法が挙げられる。文献 [58] はトランジスタを1個のシンボルで表現し、デザインルールに対してテクノロジー独立を実現する。しかし、この方法では、ゲートの折曲げや、トランジスタ上からのメタル配線の引き出し位置の自由度に対応できない。SPARCS コンパクト [59] では、トランジスタ上の配線禁止領域と端子領域をそれぞれ、ポリゴン図形で表現された protection frame (配線禁止領域)、terminal frame (端子領域) によって表現し、トランジスタやコンタクトはセルとして実現する方法が示されている。しかしながら、トランジスタ上の各々の図形要素の位置・形状が固定化されているため、レイアウト合成で課題となる図形の移動や最適化処理は出来ない。

図 6.1 に、(a) 非サリサイドプロセスにおけるセルレイアウト例と、(b) サリサイドプロセスにおけるセルレイアウトの違いを示す。図 6.1(a) では、拡散上に大きなコンタクトが打たれているが、図 6.1(b) では、十分小さなコンタクトが打たれるのみであり、拡散領域の中でコンタクトを打つ位置に関しての自由度が増大している。

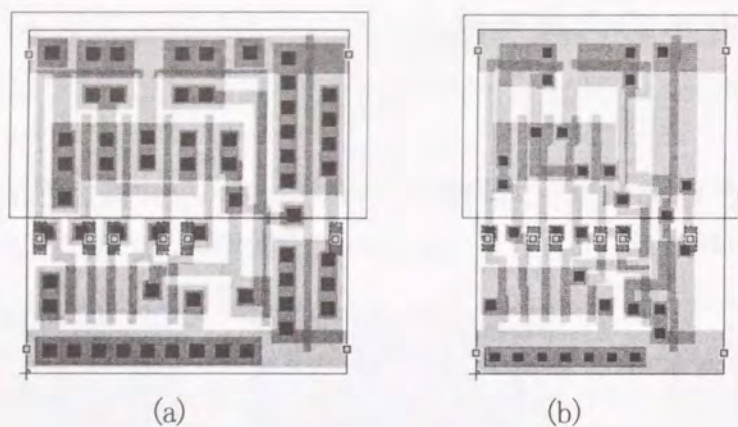


図 6.1 トランジスタのレイアウト例

このようなプロセス・テクノロジーのバリエーションを吸収するためには、例えば、次のような構造変化を許容する必要がある。

- (i) 拡散領域の形状に自由度がある。
- (ii) ゲートの45°折曲げが可能である。
- (iii) トランジスタのソース及びドレイン上から信号を引き出すためのコンタクト・パッドの大きさ及び位置に自由度がある。

本論文では、トランジスタと配線に関する柔軟性の高いレイアウトモデルと、レイアウト再生成の手法を提案する。本手法の効果が顕著に表れる例として、図 6.2 に従来のコンパクションシステムによる結果との比較を示す。コンパクションシステムは、配線を図形要素の一つとして扱い、できるだけ周りとの位置関係を保存しようとするため、不要な遠回りの配線を生じることがある。本手法によれば、配線を一旦抽象化し、配置要素のコンパクション処理と同時に配線を再生成し直すので、良好な配線結果を得ることができる。

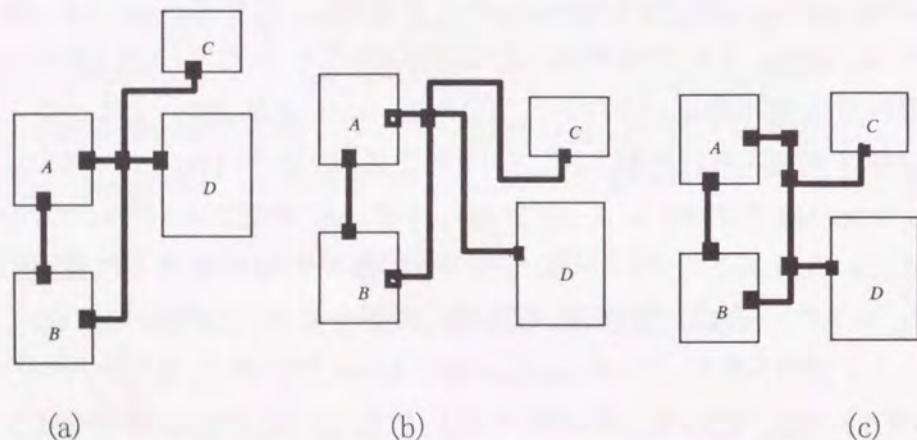


図 6.2 従来手法とのレイアウト合成結果の違い。(a) 元データの例、(b) 従来のコンパクションによるレイアウト再生成結果例、(c) 本手法によるレイアウト再生成結果例

## 6.2 テクノロジマイグレーション処理フロー

本論文で示すテクノロジマイグレーションの手法は、主要な2つの処理より構成される。一つは、与えられたレイアウトデータを一旦、テクノロジフリーな抽象化されたデータに変換する処理である。もう一方は、このテクノロジフリーデータに基づき、ターゲットプロセスに合わせたレイアウトを再合成する処理である。抽象化されたテクノロジフリーデータは、トランジスタのレイアウトや配線のジョグ等に関して十分柔軟性をもったものである。以下の節では、本論文が提案するレイアウト抽象化モデルについて述べた後、レイアウト再合成のアルゴリズムの説明を行い、最後に実際のライブラリセルへの適用例によりその効果を示す。

## 6.3 レイアウト抽象化モデル

トランジスタや配線コンタクト等のレイアウトの配置要素の図形抽象化モデルと、配置領域および配線の表現モデルについて示す。

### 6.3.1 配置要素の抽象化モデル

トランジスタレベルのレイアウト合成において、(1)トランジスタ上のメタル第1層引き出しコンタクトが、第1層メタル配線に対して配線禁止領域となること、(2)同コンタクトの位置及び大きさの最適化が求められること、(3)トランジスタ上の同コンタクト以外の領域には第1層メタル配線の通過が認められること、の理由により、トランジスタのレイアウトモデルとしては、図 6.3 に示すようにトランジスタ拡散領域に対応する矩形と、同矩形に含まれるコンタクトに対応した矩形によって表現する。尚、トランジスタ拡散領域の外形及び、コンタクトの大きさ、位置等は、レイアウト合成処理の中で順次最適化される。

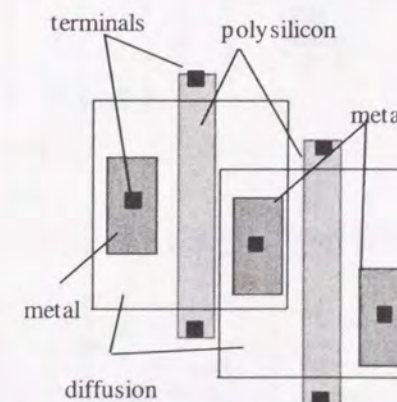


図 6.3 トランジスタの図形アブストラクション

トランジスタ拡散領域の矩形の辺上には、ゲートの端点となるポリシリコン層の端子を備える。コンタクトに対応した矩形には、同矩形の中央にメタル第1層への端子を備える。ピンと、コンタクトのアブストラクションは、1個の矩形で表現し、その中央に端子を与える。

拡散領域の共有化されたトランジスタの集合は、拡散島という概念で呼ばれる。拡散島のレ

アウト合成に関しては、ゲート折曲げを行い、拡散島上を通過するメタル第1層配線を同時に左詰めし、拡散島部分のレイアウトを合成する拡散島レイアウト生成システムをサブシステムとして用いる。

### 6.3.2 配線の抽象化モデル

本手法は、スピットとネットターゲットという配線抽象化のための新しいデータ構造を導入する。図 6.4 (a) に従来から用いられているシンボリック配線を示し、図 6.4 (b) に同配線をスピットとネットターゲットで抽象化した例を示す。スピットは、セル上に複数配置される垂直線であり、同領域に配置された矩形と、水平配線を縦に串刺しにするデータ構造である。スピット上の矩形及び配線交点は同スピット上で順番付けがされている。ネットターゲットは、配線の部分を表現する点として定義し、1) 矩形の端子上、2) 配線分岐点上、3) スピットを横切る点、の各場所に生成する。各々のネットターゲットをリンクする情報によって、配線経路と矩形との位置関係を表現する。

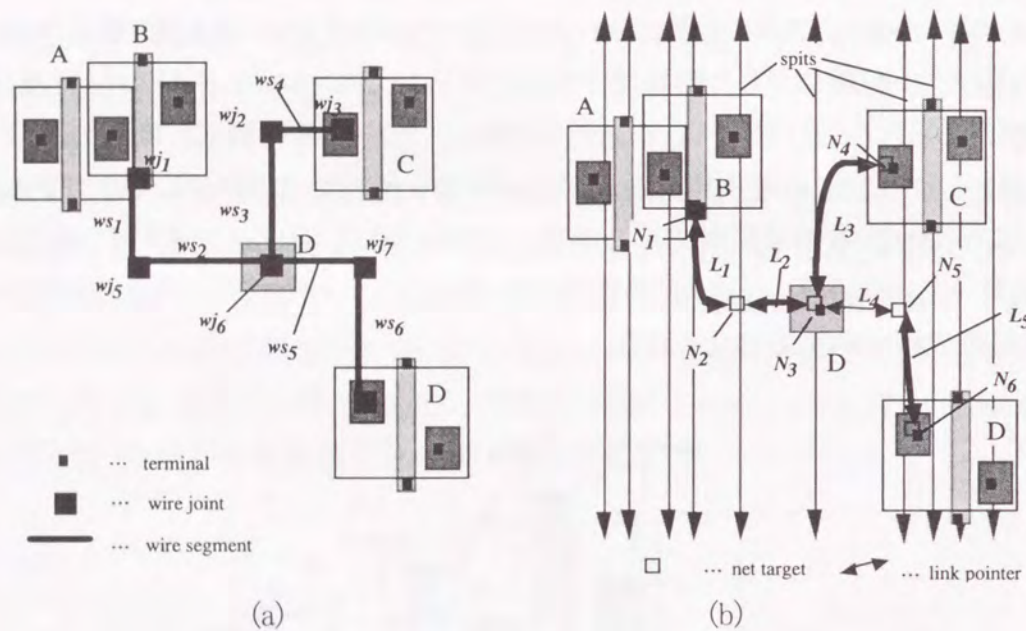


図 6.4 スピットとネットターゲット

### 6.4 レイアウト再合成処理概要

抽象化された配置要素の各矩形の垂直方向の位置制約を表現するために、図 6.5 に示すような制約グラフを用いる。同グラフの頂点は、各々の矩形に対応しており、頂点間の枝には、対応する矩形間に同領域を横切る配線を引くためのスペースを与える。この処理は前述の各スピットで隣接する矩形の間に存在する配線(ネットターゲット)を見ることによって行う。

レイアウト再合成処理においては、最初の段階に、この上下方向の配置制約グラフを用いて、最小経路法により、各々の配置要素の垂直方向の位置決めを行う。その後、スキャンラインを

用いて、水平方向のコンパクション、配線の再生成、拡散島トランジスタの図形生成を同時進行させる手法により、最終レイアウトを生成する。図 6.6 に水平スキャンによるレイアウト合成処理の概要を示す。

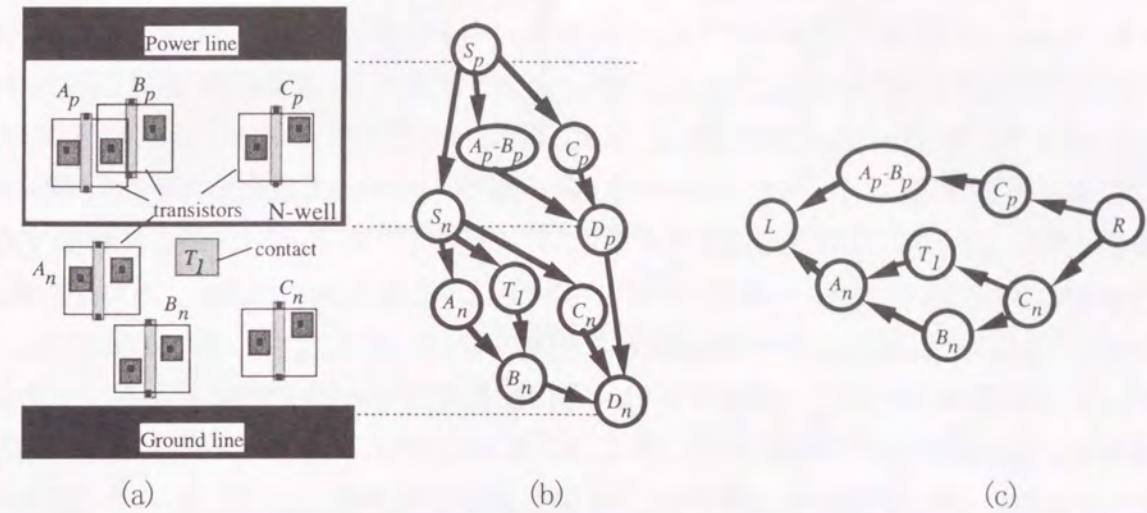


図 6.5 制約グラフ

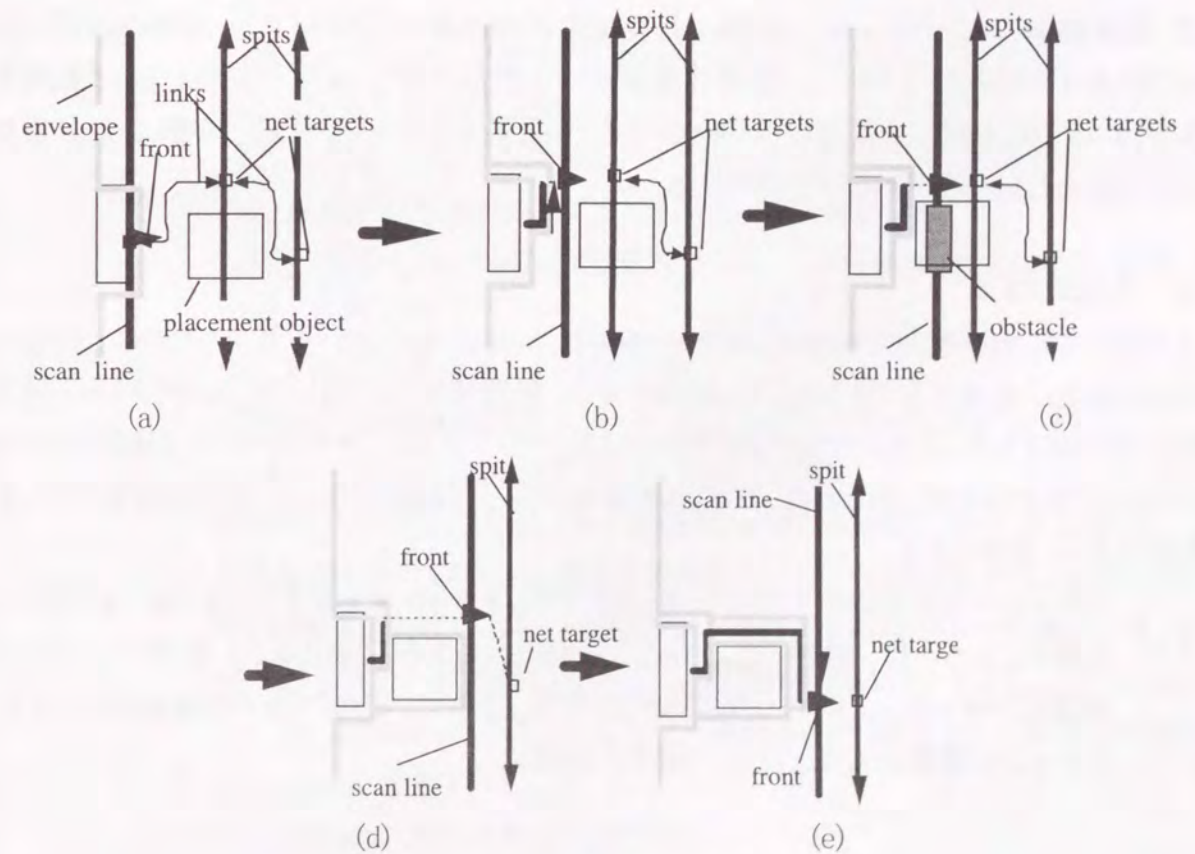


図 6.6 処理フロー概要

図 6.6(a) では、配線がネットターゲットとそれらの接続関係によって表現されている。この水平スキャン処理は、スキャンラインを左から右へ移動させながら、前述の全ての同時処理を行っていく方法である。すべての配線層に対して、包括線を定義し、スキャンし終わった配置要素や配線をすべて包括する。配線の先端を表現するフロントというデータ構造を用いる。フロントはスキャンライン上に位置し、スキャンラインの右移動に伴い、右側で一番近い同ネットのネットターゲットにできるだけ近づくようにスキャンライン上を上下動する。実際の配線は、フロントが移動したときにその軌跡となって実現される。まず、最初の段階では、スキャンラインと全ての包括線は左端に位置する。フロントは、スキャンラインが端子をスキャンしたときに発生し、同ネットの配線をし終わったときに消滅する。図 6.6 (b)に示すように、スキャンラインが右に移動するとき、フロントはできるだけつぎのネットターゲットに近づくように上方方向に移動し、フロントの軌跡を配線として生成し、包括線によって生成された配線をカバーする。次に、図 6.6(c) では、スキャンラインは配置オブジェクト左端をスキャンし、同配置オブジェクトを左詰めする。このとき、配置要素は、包括線の所まで移動できる。次に、図 6.6(d) では、スキャンラインは配置要素の左端をスキャンし、同配置要素を包括するように包括線を更新する。この間、フロントは同配置要素に邪魔されているので、つぎのネットターゲットに近づく動きはできない。次に、図 6.6(e) では、配置要素の障害が無くなり、スキャンライン上を下方方向に移動可能となるので、次のネットターゲットに向かって移動し、その軌跡を配線として生成する。以上示したように、本手法によれば、配置要素のコンパクションと配線の最生成を同時に進めることができ、必要な場所に配線の折れ曲がりが発生できる。尚、配置要素が単一の矩形でなく、複数の矩形からなる拡散島の場合、には、図 6.6(c) と同様に、スキャンラインが拡散島をスキャンすると同時に、同拡散島の図形生成を行い、同図形をを左詰めする。

## 6.5 アルゴリズム

本処理では、概略配線と同様に、矩形の移動と、詳細配線の実行を進めるために、領域の左から右方向に走査するスキャンラインを設定する。配置領域の左端には、コンパクション及び配線処理が終了したことを示すバウンダリを設ける。バウンダリは、スキャンラインが矩形の右側をスキャン仕終わった時、あるいは、フロントの移動に伴って配線が生じた場合に同矩形や、配線を含むように更新される。

1. スキャンするためのポイントリストを作成する。スキャンするポイントは、端子の存在する場所、あるいは、矩形の右あるいは左端の点である。これらを、x 座標でソートした順番にスキニングポイントリストを作成する。途中で、矩形や端子の移動があっても、スキャンの順番は変化しない。(図 6.7 参照)

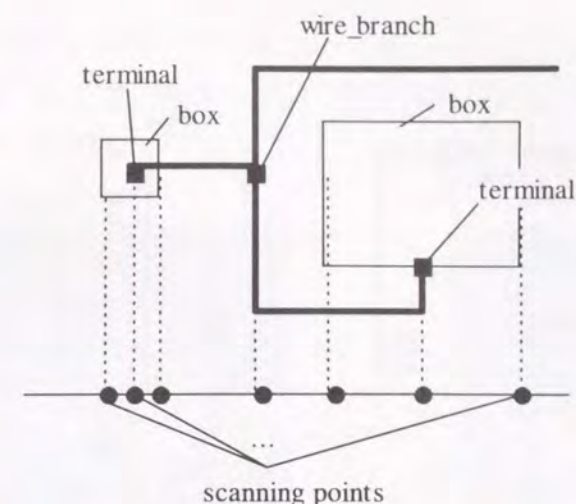


図 6.7 スキニングポイントリストの作成

2. スキニングポイントリストに従って、順次スキャンラインを移動しながら以下を行う。
  - 2.1. スキャンライン上の配線フロントを、デザインルールを満足する範囲で、できるだけターゲットに近づく方向に移動する。フロントの移動に伴い、配線を行う。
  - 2.2. スキャンラインが、スキャンする場所に応じて以下の処理を行う。
    - (A) 端子をスキャンした時は、
      - ・新しい配線フロントの生成を行い、隣接する同ネットの配線フロントと結合させ、配線を左端のバウンダリに積み上げる処理を行う。
      - ・結合できなかった浮動の配線フロントは右側へ延ばすことにより、以後の処理での配線を待つ。
    - (B) 矩形の左端をスキャンした時は、
      - ・該矩形に関する配置オブジェクトの左方向へのコンパクションを行う。
      - ・スキャンライン上に配線禁止領域を設定する。
 同矩形が拡散島の右端であった場合、拡散島生成を行い、同拡散島に含まれる矩形を全て同時に左詰めする[60]。ただし、この段階では、左端のバウンダリは更新されていない。拡散島上を通過する配線があった場合でも、拡散島生成システムが同配線を通過する領域を確保していれば、スキニングポイントによるスケジューリングに従って概略配線で指定された場所に形成される。(図 6.8 参照)
    - (C) 矩形の右端をスキャンした時は、
      - ・該矩形に関する配置オブジェクトを左端のバウンダリに積み上げる処理を行う。



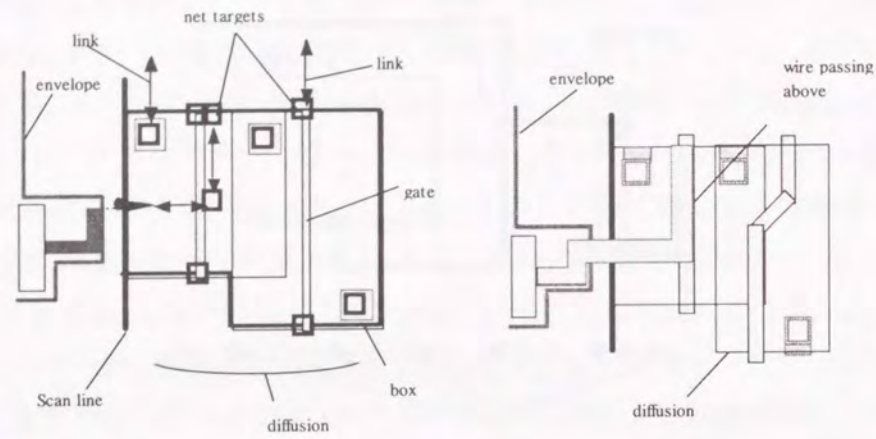
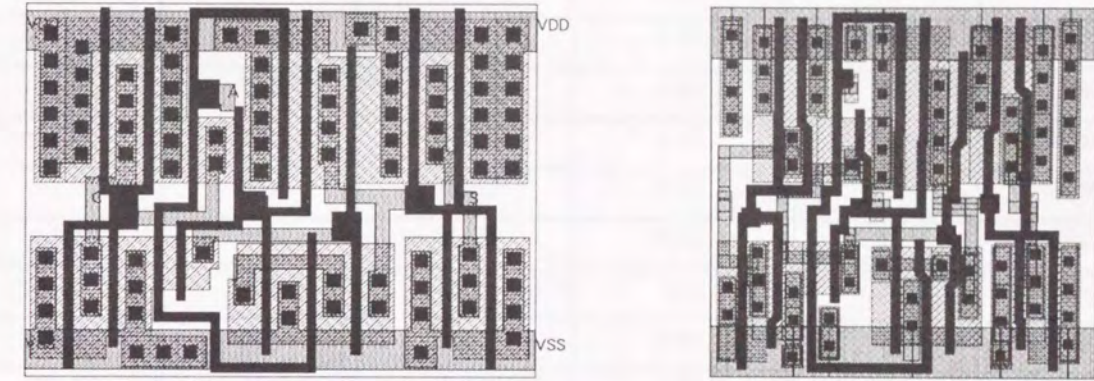


図 6.8 拡散島図形生成処理、(a) 矩形による抽象化データ、  
(b) 生成された拡散島図形

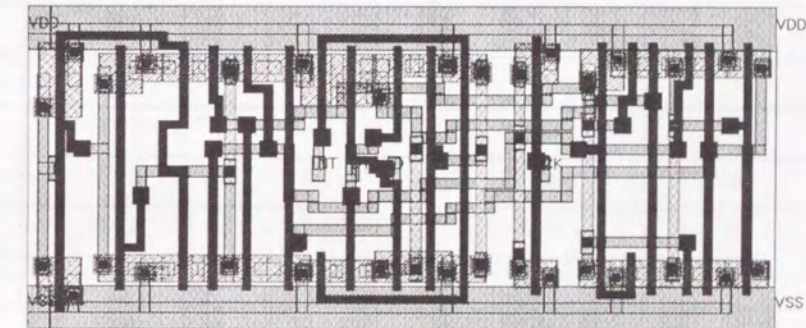
## 6.6 実験結果

図 6.9, 図 6.10 それぞれに、本システムによって合成した標準セルの例を示す。人手設計と同様の結果が得られている。また、表 6.1には、配置以降の処理を自動で行い、人手設計と比較した結果を示している。約9割のデータに関して人手並みの結果が得られたことを示している。

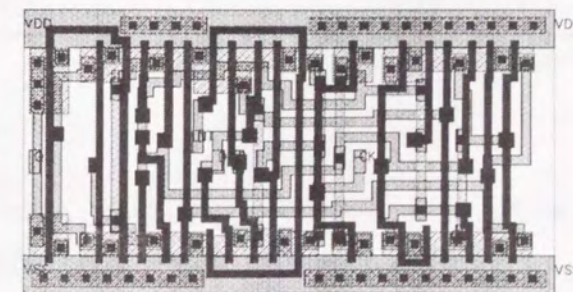


(a) (b)

図 6.9 テクノロジマイグレーションの適用例 (a) 変換前、(b)変換後



(a)



(b)

図 6.10 自動合成における適用例 (a) 配線処理後、(b)再生成後

表 6.1 人手設計との比較

セル名	トランジスタ数	セル幅(*1) (本手法)	セル幅(*2) (人手設計)	人手比 (%) (*1/*2)	計算時間(sec)
NAND3	9	16.8	16.8	100	0.18
NAND6A	18	24.0	24.0	100	0.43
NAND6B	18	24.0	24.0	100	0.47
OR2	6	10.3	9.6	107	0.17
OR4B	10	14.4	14.4	100	0.27
OR5B	12	24.0	24.0	100	0.25
NOR2	6	12.0	12.0	100	0.12
NOR3A	6	9.6	9.6	100	0.15
NOR3B	9	16.8	16.8	100	0.22
NOR5A	10	14.4	14.4	100	0.27
NOR5B	14	19.2	19.2	100	0.33
NOR6A	20	24.0	24.0	100	0.43
NOR6B	20	26.6	24.0	111	0.63
BUF1	8	15.9	14.4	110	0.18
BUF2C	25	34.4	31.2	112	0.54
BUF7	8	12.0	12.0	100	0.18
BUF8	8	12.0	12.0	100	0.22
INV1	18	24.4	24.0	102	0.32
XOR2	16	23.8	19.2	124	0.33
XOR3	26	37.2	36.0	103	0.68
FF1	23	31.2	31.2	100	0.57
FF2	26	36.1	36.0	100	0.73
FF3	38	59.6	55.2	108	1.17

注)\*1, \*2, それぞれのセル幅のユニットは設計グリッドを用いている。

## 6.7 結言

本文では、トランジスタに対する新しいレイアウトモデルを示し、プロセスの変化に容易に追従出来ることを示した。同レイアウトモデルを用いて、柔軟性の高い概略配線方法と、配線とコンパクションを同時進行させることによって、レイアウトを実現するトランジスタレベルの新しいレイアウト生成方法を示した。また、実験により、レイアウト合成結果が人手設計並みの良好な結果であることも示した。本システムを、実際のライブラリ開発において適用し、約3時間で300セルのレイアウトを回路から合成、約1時間で既存セルからの流用を行うことができた。(使用マシン環境:SUN SPARKStation20、C言語によってコーディング) 合成された300セルの内、

90%は人手で1セルあたり半日かけて作られたレイアウト結果の面積に匹敵する。(残りの10%は、人手より約5~10%大きくなった。) 本システムの実用化により、セルライブラリの総開発工数は約10分の1に削減された。

## 第7章 結論

大規模集積回路のレイアウト合成に関する研究として、まず第1章では、産業的背景を述べ、大規模集積回路の規模および設計困難度が今後ますます増大するため、レイアウト合成の技術革新も意欲的に進めていく必要があることを示した。

第2章では、大規模集積回路の複数のレイアウト構造(方式)について解説し、それぞれの構造(方式)に対する課題をまとめ、本研究の意義として、(1)ゲートアレイ方式LSIのレイアウト合成において、手戻りの少ない設計フローの構築すること、(2)大規模化するフルカスタム方式LSIの設計工数と期間を短縮すると同時に、面積・遅延において良好な結果を得ること、(3)セルライブラリの新規および流用設計工数と設計期間を短縮することのそれぞれの重要性について示した。

第3章では、ゲートアレイ方式LSIの配線成功率を向上するための等価端子割り当て手法を提案し、有効性を確認した。本手法によれば、ランダムに等価端子割当を行う場合に比較して、与えられた配線トラックに対する余裕が、1～2本余計に確保できることを示した。また、詳細配線のトラック数を増加させる主な要因である上下制約グラフの上下矛盾の個数を大幅に減少することを確認した。

第4章では、大規模集積回路のフロアプランを最適化する方法を提案し、実LSIのレイアウト合成への適用により効果を実証した。配線領域の面積も考慮しながら各々のブロックの形状を精度良く最適化することを、配線チャンネルが効果的に使用されている実験例によって示した。また、ブロック外部ピン配置に関しては、ピンの分散とブロック間概略配線長の最小化を同時に解決する方法を述べ、実験によりブロック間配線の混雑度合いが緩和されることを示した。さらに、電源配線に関しては、複数の動作状態を考慮し、配線要素における電流密度の上限値とブロック端子における電圧降下の制限を満足し、かつ、チップ面積を最小化する方法について述べた。

第5章では、大規模集積回路のブロック間配線方法として、面積最小化と遅延制約の満足との両方の評価指標に対して良好な結果を得るための概略配線手法を提案した。実験により、配線遅延対遅延制約の比の最大値が、111%(障害あり)から、82%(全遅延制約を満足)まで減らすことを確認した。また、L字チャンネルを使った配線領域分割方法を提案し、実LSIの配置配線により有効性を実証した。

第6章では、標準セルの合成方法として、トランジスタに対する新しいレイアウトモデルを提案し、同モデルを用いて柔軟性の高い配線、コンパクション、と拡散島図形生成を同時に進行させる方法を提案し、実セルライブラリへの適用により有効性を確認した。ライブラリシリーズの約300セルに対して実験を行い、90%のセルに関しては、レベルの高い設計者が人手で約半日/セルかけて設計した面積と同等であった。残りの10%に対しては、人手設計より5%～10%大きくなったが、設計者に対して良好な初期解をあたえることにより、実ライブラリ開発のトータ

ルの設計工数を約1/10に削減し実用面でも大きな効果を示した。

今後は、図7.1に示すように、設計規模がますます大規模化する様相を示している。設計上流でのHDL等の記述言語の使用をより普及し、設計抽象度を上げて行くことに加えて、設計の再利用化を推し進める必要がある。レイアウト設計についても有効性の高いテクノロジー変換技術の確立と実用化を進めることが課題として上げられる。

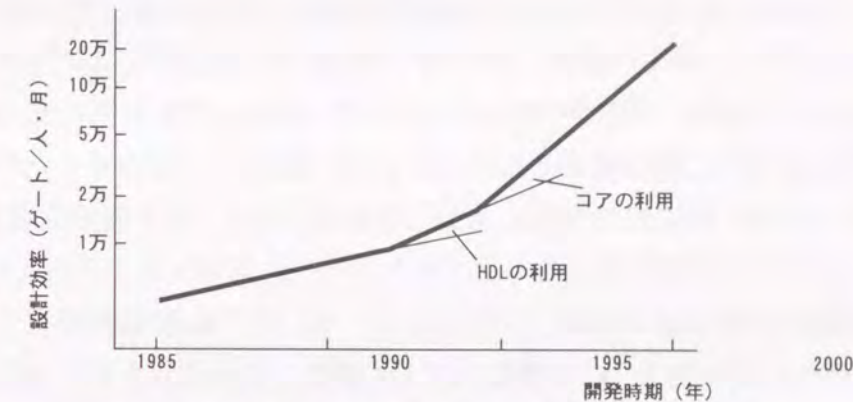


図7.1 設計の大規模化とそれを支える設計戦略

次に、微細化に伴って、ノイズや製造ばらつきの回路およびレイアウトに対する影響が無視できないものとなってきており、微細化物理現象を考慮したレイアウト最適化も重要となる。さらに設計抽象度の上位化に伴い、ハイレベル合成やハード・ソフト・コデザインといったレベルでも、レイアウトイメージを与え遅延等に対する推定を可能とする上流設計用のフロアプラン技術の確立も重要と考える。(図7.2)

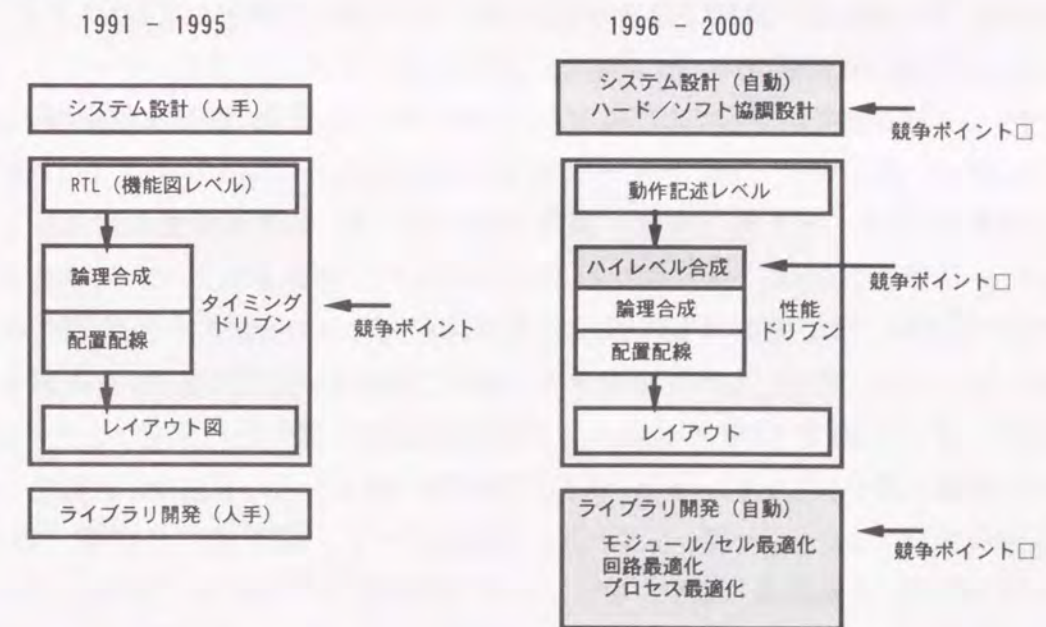


図7.2 設計抽象度の上流へのシフトと競争ポイントの推移

## 謝辞

本論文を執筆するにあたり多数の方々のお世話になりましたので、ここに記して感謝の意を表します。

大阪大学在学中に御指導を賜るとともに、本論文の作成にあたり終始懇切なご指導とご教示を賜りました大阪大学工学研究科情報システム工学専攻 白川功教授に厚く感謝を申し上げます。さらに本論文をまとめるにあたり、有益なご指導とご助言を賜った大阪大学工学研究科情報システム工学専攻 藤岡弘教授、村上孝三教授に厚く感謝申し上げます。

大阪大学においてご指導、ご鞭撻を賜った大阪大学 尾崎弘名誉教授に深く感謝致します。大阪大学在学中に直接のご指導を賜るとともに、つねに励ましとご助言をいただく中央大学理工学部電気電子工学科 築山修治教授に心から感謝致します。松下電器産業株式会社半導体研究センターにおいて研究の機会を与えていただくとともに多大なご支援をいただきました現 松下電器産業株式会社特別顧問 水野博之氏、現 松下電器産業株式会社顧問 堀内司朗氏、現 研究本部副本部長 竹本豊樹氏に深く感謝の意を表します。また、半導体開発本部において本研究に対してご援助とご指導をいただきます半導体開発本部部長 古池進氏、半導体開発本部半導体先行開発センター所長 西嶋修氏に深く感謝致します。さらに松下電器産業株式会社での研究生活のほぼ全般に渡って、あたたかくご指導とご鞭撻いただいた現 松下電器産業株式会社半導体開発本部マイコン開発センター所長 間野洋治郎氏、現 近畿大学生物理工学部電子システム情報工学科 秋濃俊郎教授に深謝致します。

本論文の第3章における研究は、大阪大学において、現 日本電信電話株式会社 原田育生氏、現 株式会社東芝 池田敏雄氏、現 日本電気株式会社 木本務氏、および現 シャープ株式会社 竹田信弘氏と共同で進めたものです。ここに記して感謝の意を表します。

第4章および第5章の研究は、松下電器産業株式会社半導体研究センターにおいて行ったものです。システム開発に際してのご協力とご討議を頂きました現 研究本部医療画像情報開発チームリーダー 羽山繁氏、現 半導体開発本部半導体先行開発センター 山口龍一主任技師、山本敦志技師、川上善之技師、蕪尾知恵技師、現 半導体開発本部 CE システム開発センター 田中康弘技師、株式会社松下ソフトリサーチ 塩原孝弘リーダー に感謝致します。

第6章の研究は、松下電器産業株式会社半導体研究センターおよび半導体開発本部において進めたものであります。本研究に対するご指導ご鞭撻をいただく現 半導体開発本部マイコン開発センター主担当 国信茂郎氏、半導体開発本部半導体先行開発センター主担当 松澤昭氏に感謝の意を表します。また、ライブラリ開発の観点からの有益なご討論をいただいた半導体先行開発センター 大谷一弘リーダー、岡晶久リーダー、藤原睦主任技師、谷口博樹技師、石坂高志氏 に感謝致します。さらにシステム開発へのご協力とご討議をいただいた半導体先行開発センター 雑賀俊二技師、四宮典子技師、田中正和技師、株式会社松下ソフトリサーチ 森下直人技師に謝意を表します。

最後に、日頃より有意義なご助言、ご討論いただく半導体開発本部の皆様に感謝の意を表します。

なお、第3章の研究の一部は、文部省科学研究費補助金：一般研究 B57460117(昭和57、58年度)の援助のもとに行われたものです。

## 参考文献

- [1] SIA, "The national technology roadmap for semiconductors: Technology Needs," SEMATEC (1997).
- [2] G. Persky, D. Deutsch, and D. Schweikert, "LTX-A system for directed automatic design of LSI circuits," in *Proc. Design Automation Conference*, pp. 399-407 (1976).
- [3] 山本敦志、福井正博、羽山繁、間野洋治郎, "シミュレティッドアニーリング法を用いたスタンダードセル自動配置の一手法," 信学技報, CAS86-191, pp. 25-31 (1987).
- [4] D. Dura and T. Tsang, "A sub-micron ASIC low power static RAM compiler," in *Proc. International Conference of Computer-Aided Design*, pp. 226-230 (1996).
- [5] 川上善之、山口龍一、福井正博、羽山繁、秋濃俊郎, "論理回路の規則的繰り返しが多く存在するブロックにおけるセル配置の一手法," 信学技報, VLD89-48 pp. 73-79 (1989).
- [6] T. Adachi, H. Kitazawa, M. Nagatani, and T. Sudo, "Hierarchical top down layout design method for VLSI chip," in *Proc. Design Automation Conference*, pp. 785-791 (1982).
- [7] M. Y. Hsueh and D. O. Pederson, "Computer-aided layout of LSI circuit building-blocks," in *Proc. International Symposium on Circuits and Systems*, pp. 474-477 (1979).
- [8] S. Kimura, N. Kubo, T. Chiba, and I. Nishioka, "An automatic routing scheme for general cell LSI," *IEEE Transactions on Computer-Aided Design of Integrated Circuits And Systems*, vol. CAD-2, no. 4, pp. 285-291 (1983).
- [9] W. M. Dai, T. Asano, and E. S. Kuh, "Routing region definition and ordering scheme for building block layout," *IEEE Transactions on Computer-Aided Design of Integrated Circuits And Systems*, Vol. CAD-3, no. 3, pp. 218-225 (1984).
- [10] Barbara T. : "Make-or-buy library decision faces COT/foundry customers," *Computer Design* (1997).
- [11] 瀧和男, "パストランジスタ論理に関する動向調査報告," 情処学 DA シンポジウム, pp. 147-154 (1997).
- [12] M. Fukui and A. R. Newton, "Optimum module generation for semi-custom," in *Proc. IEEE Asia-Pacific Conference on Circuits and Systems*, pp. 184-189 (1992).
- [13] J. Fishburn and A. Dunlop, "TILOS: A posynomial programming approach to Transactions istor sizing," in *Proc. International Conference on Computer-Aided Design*, pp. 326-328 (1985).
- [14] M. Tanaka, M. Fukui, and S. Kuninobu, "A precise estimation model of cell area and Transactions istor capacitance for Transactions istor size optimization," in *Proc. Synthesis and System Integration of Mixed Technologies*, pp. 241-247 (1998).
- [15] T. Uehara and W.M. VanCleemput, "Optimal layout of CMOS functional arrays," *IEEE Transactions Computer*, vol.-30, pp. 305-312 (1981).
- [16] S. Saika, M. Fukui, N. Shinomiya, and T. Akino, "A two-dimensional placement algorithm for cell synthesis and its application to standard cells," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E80-A, No.10, pp. 1883-1891 (1997).
- [17] M. Fukui, N. Shinomiya, and T. Akino, "A new layout synthesis for leaf cell design," in *Proc. Asia and South Pacific Design Automation Conference*, pp. 259-264 (1995).
- [18] H. Shin and A. Sangiovanni-Vincentelli, "MIGHTY: A 'rip-up and re-route' detailed router," in *Proc. International Conference on Computer Aided Design*, pp. 2-5 (1986).
- [19] 四宮典子、福井正博、西垣泰男, "リーフセル用配線システム," 信学技報, CPSY97-71, pp. 21-26. (1997年).
- [20] H. Shin and A. Sangiovanni-Vincentelli, "Two dimensional compaction by 'Zone Refining'," in *Proc. of Design Automation Conference*, pp. 115-119 (1986).
- [21] W. Wolf, J. Newkirk, and R. Dutton, "Two-dimensional compaction strategies," *IEEE International Conference on Computer Aided Design*, pp. 90-91 (1983).
- [22] 福井正博、築山修治、白川功, "ゲートアレイ方式 LSI の端子割り当て問題に対する一手法," 電子情報通信学会論文誌(C), Vol. J66-C, No. 12, pp. 1172-1179 (1983年).
- [23] M. Fukui, A. Yamamoto, R. Yamaguchi, and S. Hayama, "SMILE- a hierarchical layout system for building block LSI," *International Journal of Computer Aided VLSI Design*, Vol. 1, No. 3, pp. 281-302 (1989).
- [24] T. Shiohara, and M. Fukui, "A pin assignment and global routing algorithm for floorplanning," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E81-A, No.8 (1998).
- [25] M. Fukui, Y. Tanaka, and T. Akino, "An algorithm for power and ground routing in building block VLSI," in *Proc. Synthesis and Simulation Meeting and International Interchange*, pp. 294-299 (1990).
- [26] M. Fukui, A. Yamamoto, R. Yamaguchi, S. Hayama, and Y. Mano, "A block interconnection algorithm for hierarchical layout system," *IEEE Transactions actions on Computer-Aided Design of Integrated Circuits And Systems*, Vol.CAD-6, No.3, pp. 383-391 (1987).
- [27] M. Fukui, N. Shinomiya, S. Saika, T. Akino, and S. Kuninobu, "Layout abstraction and technology retargeting for leaf cells," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E81-A, No.12 (1998) to be published.
- [28] R. Kamikawa, K. Kishida, A. Osawa, I. Yasuda, and T. Chiba, "Placement and routing program for masterslice LSI's," in *Proc. Design Automation Conference*, pp. 245-250 (1976).
- [29] K. Ueda, Y. Sugiyama, and K. Wada, "An automated layout system for masterslice LSI: MARK," *IEEE Journal of Solid-State Circuits*, SC-13,5, pp. 716-21(1978).

- [30] T. Yoshimura and E. S. Kuh, "Efficient algorithms for channel routing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits And Systems*, pp. 25-36 (1982).
- [31] 福井正博、築山修治、白川功、尾崎弘、"チャンネル配線の一手法," 信学技報、CAS81-121, pp. 91-98 (1981年2月).
- [32] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM Journal Computer*, 1, 1, pp. 146-160 (1972).
- [33] 野田、寺井、佐藤、八原、"多様なデバイスのゲートアレイに適用可能な配線プログラム," 信学技報、CAS82-79 (1982).
- [34] R. Otten, "Automatic floor-plan design," in *Proc. Design Automation Conference.*, pp. 261-267 (1982).
- [35] K. Maling, S. H. Muller, and W. R. Heller, "On finding most optimal rectangular package plans," in *Proc. of Design Automation Conference*, pp. 663-670 (1982).
- [36] S. Tsukiyama, K. Koike, and I. Shirakawa, "An algorithm to eliminate all complex triangles in a maximal planar graph for use in VLSI floor-plan," in *Proc. International Symposium on Circuits and Systems*, pp. 321-324 (1986).
- [37] D. Wong and C. Lin, "A new algorithm for floor-plan design," in *Proc. Design Automation Conference*, pp. 101-107 (1986).
- [38] W. M. Dai and E. S. Kuh, "Simultaneous floor-planning and global routing for hierarchical building block layout," in *Proc. International Conference on Computer Aided Design*, pp. 334-340 (1986).
- [39] K. Ueda, "Placement algorithm for logic modules," *IEE Electronics Letter*, 10, 10, pp. 206-208 (1974).
- [40] 山口龍一、福井正博、山本敦志、羽山繁、"階層的レイアウトシステムにおけるスタンダードセル自動配線の一手法," 情報処理学会第34回全国大会, pp. 1969-1970 (1987年10月).
- [41] 山本敦志、福井正博、山口龍一、羽山繁、"階層的レイアウトシステムにおけるスタンダードセル自動配置の一手法," 情報処理学会第34回全国大会, pp. 1991-1992 (1987年10月).
- [42] M. Pedram, M. Marek-Sadowska, and E.S.Kuh, "Floorplanning with pin assignment," in *Proc. International Conference on Computer Aided Design*, pp.98-101 (1990).
- [43] E. L. Lawler, "Combinatorial optimization: networks and matroids," *Holt, Rinehart and Wilson*, N. Y. (1976).
- [44] S. Chowdhury and M. A. Brewer, "Minimal area sizing of power and ground nets for VLSI circuits," in *Proc. MIT Conference on Advanced Research in VLSI* (1986).
- [45] S. Chowdhury, "Optimum design of reliable IC power networks having general graph topologies," in *Proc. Design Automation Conference*, pp. 787-790 (1989).
- [46] S. Haruyama, and D. Fussell, "A new area-efficient power routing algorithm for VLSI

- layout," in *Proc. International Conference on Computer Aided Design*, pp. 1123-1129 (1985).
- [47] R. Fletcher, "Practical methods of optimization," Vol. 2, John Wiley & Sons (1981).
- [48] L. Stockmayer, "Optimal orientations of cells in slicing floorplan designs," *Information and Control*, Vol. 59, pp. 91-101 (1983).
- [49] 福井正博、岩崎知恵、羽山繁、"チャンネル構造列挙の一手法," 情処研報、DA44-10, pp. 75-82 (1988).
- [50] 福井正博、山本敦志、岩崎知恵、羽山繁、"ブロック形状最適化の一手法," 電子情報通信学会秋季全国大会, pp. A-1-163~164 (1988).
- [51] T. Yoshimura and E. S. Kuh, "Efficient Algorithms for Channel Routing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits And Systems*, pp. 25-36 (1982).
- [52] T. G. Hamachi and J. K. Ousterhout, "A Switch-box Router with Obstacle Avoidance," in *Proc. Design Automation Conference*, pp. 173-179 (1984).
- [53] M. Fukui, N. Shinomiya, and T. Akino, "A new layout synthesis for leaf cell design," in *Proc. Asia and South Pacific Design Automation Conference*, pp. 259-264 (1995).
- [54] C. Meed and L. Conway, "Introduction to VLSI Systems," Addison Wesley (1979).
- [55] D. Gibson and S. Nance, "SLIC - Symbolic Layout of Integrated Circuits," in *Proc. Design Automation Conference*, pp. 434-440 (1976).
- [56] B. Preas and M. Lorenzetti, "Physical Design Automation of VLSI Systems," Chapter 6, The Benjamin/Cummings Publishing Co. (1988).
- [57] P. Kollaritsch and N. Weste, "TOPOLOGIZER," *IEEE Journal Solid-State Circuits*, vol. sc-20, No.3, pp. 799-803 (1985).
- [58] P. Smith and S. Daniel, "The VIVID System approach to technology Independence," in *Proc. of Design Automation Conference*, pp. 115-122 (1984).
- [59] J. Burns and A.R. Newton, "SPARKS: A New Constraint-Based IC Symbolic Layout Spacer," in *Proc. Custom Integulated Circuit Conference*, pp. 534-539 (1986).
- [60] 四宮典子、福井正博、雑賀俊二、秋濃俊郎、"屈曲ゲートを用いたレイアウト最適化手法," 信学技報、VLD95-155, pp. 97-104 (1996年3月).

