

## アナログマスタースライス自動レイアウトシステム

上高原 道夫 永原 出 篠崎 圭一 熊王 尚子 秋山 史子 井上 明美

ソニー株式会社 セミコンダクタカンパニー  
〒243 神奈川県厚木市旭町4-14-1 厚木テクノロジーセンター  
michio@saskg.semicon.sony.co.jp

本レポートでは、アナログマスタースライス用の新しいレイアウト設計手法について報告する。アナログLSI特有のレイアウト設計制約は、アナログLSIの特性を左右する重要な設計上の項目であるが、既存の自動化手法では、これらの設計制約を十分に解決する事が困難であった。我々のシステムでは人手設計と自動設計の長所をシステム上で融合させ、個別の設計制約には人手設計で設計者の意図を忠実にレイアウトへ反映させ、その他は自動設計手法が利用できるようにしている。また、本システム用に開発した素子レベル自動配線は、線分展開法をベースにアナログマスタースライス特有のレイアウトモデルに対応させたものである。本システムによって、設計工数を従来手法の1/3に削減する事が出来た。

## AN ANALOG MASTER SLICE AUTOMATIC LAYOUT SYSTEM

Michio Kamitakahara, Izuru Nagahara, Keiichi Shinozaki, Naoko Kumaoh,  
Fumiko Akiyama and Akemi Inoue

Semiconductor Company, Sony Corporation  
Atsugi Technology Centor 4-14-1, Asahi-chou, Atsugi-shi, Kanagawa-ken, 243 JAPAN

In this paper, we present a new design methodology for analog master slice LSIs. Analog layout requirements are very important for LSIs because it determines the character of the analog LSIs. However, analog layout requirements are not easily automated with the current design methodology. We developed a system based on the concept of a flexible integration of manual and automatic designing methods. Placement and analog constraint path are designed with manual designing methods and the remaining tasks are completed with automatic designing methods. A flexible device level router has been developed for analog master slice. It is based on a line-expansion algorithm and we improved it for analog layout. Compared with the existing conventional systems, it is possible to reduce the design time drastically to 1/3 for complex analog master slice designs.

## 1. はじめに

アナログマスタースライスとは、図1のように予め下地上に敷き詰められた基本素子（トランジスタ、抵抗、容量等）を使用し、その素子間を配線することにより、LSIレイアウトを実現する手法である。よって、レイアウト及びプロセスにおけるカスタマイズ作業はアルミ配線のみであり、LSI開発工数がカスタムLSIに比べて（特にプロセスにおいて）短いという特徴を持つ。

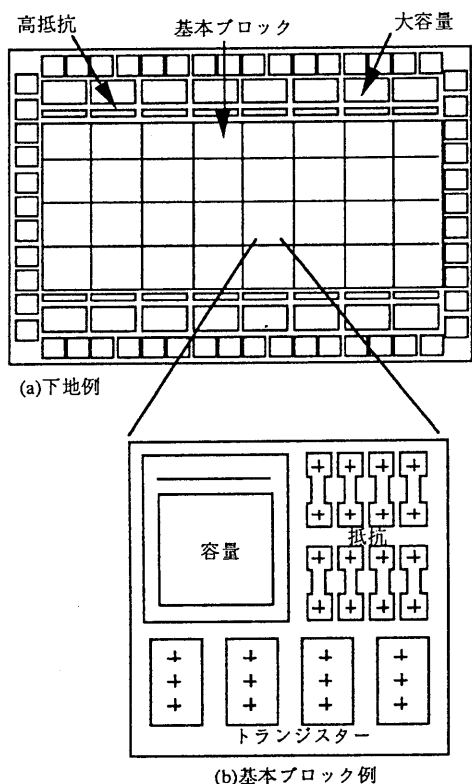


図1. マスタースライス下地例

設計自動化ツールの普及により、デジタルLSIレイアウトの自動化が進む一方で、アナログLSIのレイアウトは、その特有の制約から自動化が困難であり、筆者

等の知る限りにおいて、バイポーラアナログマスタースライス用レイアウトの自動化報告はなされておらず、これまで人手による設計が主体であった。

本システムは、2層アルミ配線のバイポーラプロセスにおけるアナログマスタースライスLSIの実用レイアウトシステムとして、設計工数を大幅に削減する事を目的に開発を行った。

システムを開発するにあたり、現状の設計作業を分析したところ、配置/配線作業が全体のレイアウト設計工数に占める割合は、各々について約8%と8.7%であった。また、配線作業のうちアナログ制約部分の作業が全体のレイアウト設計工数に占める割合は約4%である。

レイアウト設計工数を大幅に削減するためには、自動レイアウト手法の導入が考えられるが、配置やアナログ制約配線では、自動処理用の設定やアナログ制約を設計からレイアウトへ伝達する仕組の構築が必要である。配置やアナログ制約配線は、現状においても設計に占める作業工数は少なく、自動化する上でのオーバーヘッドやシステムを複雑化してしまう損失を考慮すると、自動化による設計工数の削減効果は極めて小さい。配置はアナログ制約指定（隣接配置、対称配置等）が多く、特性への影響が大きい。更に配置の結果が、自動配線の配線時間や配線の質、配線率に大きな影響を与える事が、経験上判明している。

これらの分析結果から、本システムにおいては、全体の作業工数に占める割合は少ないが、レイアウト品質やLSIの特性を左右する配置やアナログ制約配線を人手設計、その他の配線を自動設計とし、それらの手法をシステム上で融合させる仕組を開発する事で、今までと同等品質のLSIを少ない工数で実現する事を目指した。

## 2. 設計フロー

本システムは、図2の設計フローに示したとおり、レイアウトから検証、マスクデータ生成までの統合システムである。レイアウト作業は、まずデバイスマッピングによって、回路図をレイアウト用の図面に変換する作業から始める。

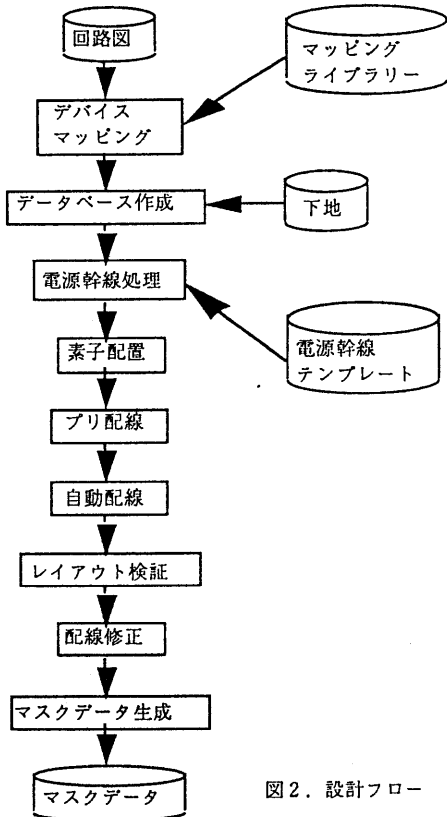


図2. 設計フロー

次に、変換された図面を階層展開し、接続情報を抽出する事により、レイアウト用データベースを作成する。以後は、このデータベース上で人手設計と自動設計を組み合わせ使用し、レイアウトを完成させる。

素子配置やプリ配線、配線修正といった人手作業においても、データベースの接続情報を積極的に利用している。素子配置では、回路図上で配置素子を選択するスキマティックドリブンレイアウト機能により、回路図上の素子位置などを参考にしたレイアウトを可能にしている。人手配線では、接続情報を保った配線修正を行うネットリストエディット機能により、同一ノードの配線や端子をハイライトする事で、作業効率の向上を図った。

以降において、本システムの特徴的な機能について説明する。

## 3. デバイスマッピング

アナログ回路では、回路を構成する素子は各々抵抗値や容量値などのパラメータを持っており、それによってレイアウト上での素子サイズが決定される。しかし、マスタースライスでは、基本素子が一定のサイズで下地上に敷き詰められており、設計者は回路図上の各パラメータをレイアウト上で実現するために、基本素子を組み合わせ回路をレイアウト上に生成する必要がある。

既存のCADにおいては図3に示すように、A)下地の基本素子のパラメータに合わせて回路図を作成する、またはB)パラメータに対応する基本素子のレイアウトパターンをマクロセルとして用意しておき、回路上の素子をパラメータ毎にマクロセルへ割り付けている。A)の方法は、回路図作成工数が増大する。B)の方法は、下地が不均一なため、下地の素子使用率を向上させるためには、一つのパラメータに対して複数のマクロセルを準備する必要があり、それをプロセス毎に用意しなければならず、システム保

守の面で負担が大きい。

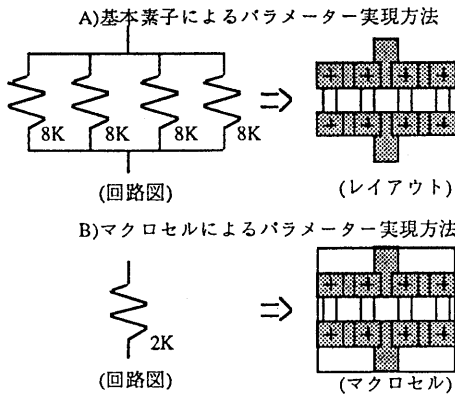


図3.既存手法による素子パラメータ対応例

本デバイスマッピング手法では、素子のパラメータに対応するレイアウトパターンをマクロセルとして準備するのではなく、パラメータに対応した回路図をマッピング素子として用意しておき、回路図上の素子をマッピング素子へ置換する。

具体的には、通常使用するパラメータを下地上の基本素子の組み合わせによって実現した回路を作成し、これをマッピング素子としてライブラリへ登録しておく。回路図のデバイスマッピング処理をする際には、回路上の素子とパラメータをキーにマッピングライブラリを検索し、回路図上の素子をマッピング素子に置換する。これによって、回路図上の素子は、基本素子からなる回路へ階層化される。図4は、下地の抵抗素子が $8\text{K}\Omega$ で、回路図の素子“R”のパラメータが $16\text{K}\Omega$ の時、デバイスマッピング処理によって、素子“R16K”に配置された例である。素子“R16K”は、下地の抵抗に対応する素子“R”、パラメータ $8\text{K}\Omega$ の抵抗を直列接続する事によって、パラメータ $16\text{K}\Omega$ を実現した回路を下位階層に持っており、これによって回路図と

レイアウトの各素子が完全に対応する。

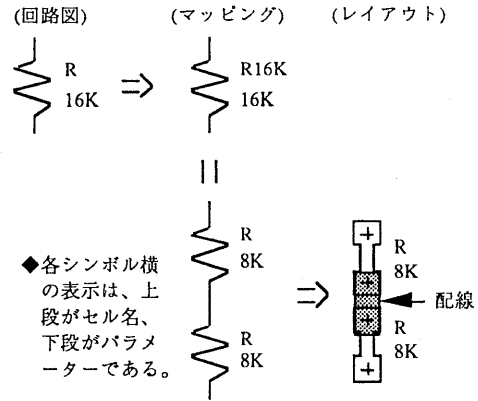


図4.デバイスマッピング例

マッピングライブラリは、基本素子のパラメータが同じであれば、異なる下地やプロセスでも利用が可能である。また、チップ固有の特殊なパラメータが必要な場合は、基本素子を組み合わせることで所望のパラメータを実現する回路図を作成すれば、本手法によって回路図の変換が可能である。

#### 4. 電源幹線処理

デジタル系マスタースライスでは、下地上の素子の位置が固定かつ電源が1種類であるため、電源配線を下地に埋め込む、または電源配線をマクロセル化する方法が一般に行われているが、アナログ回路では電源の種類が複数あり、電源配線を固定化する手法は利用できない。このような場合、既存CADでは自動配線ツールに備わっている経路指定や配線エディタ機能を使って、設計者が入力を行っているが、アナログ回路では電源配線の幅指定や、素子分離用の基盤電源配線など複雑な作業が必要であり、制約の多い配線エディタでは作業効率が悪い。

そこで本システムでは、電源配線マクロセルをレイアウト上で展開し、電源配線を必要に応じてポリゴンエディタで分断/修正した後、配線図形に対して所望のネットを割り当てる（ネットアサイン）方法をとっている。ネットアサインにおいては、ネットを割り当てる配線の一部を選択すれば、その配線と同じ電位の配線群をレイアウト上から図形演算によって自動抽出する機能があり、基盤電源配線など多くの詳細な配線を実際に選択することが可能である。

## 5. 自動配線

アナログ回路のレイアウトでは、回路に依存するアナログ制約以外に、電源の支線配線やPAD周辺、コンタクト生成においてデジタル系とは異なる処理が必要である。また、下地上にはサイズや形状、端子数が各々異なる素子が不均一に並んでいる。このように、複雑なレイアウトモデルを扱う為に、自動配線には柔軟な配線コントロール機能が必要である。本システムでは2次元ルーターである線分展開法[1]をベースに、アナログレイアウト機能を追加した。その特徴について説明する。

### 5. 1. 電源配線への一方向支線接続

電源幹線から支線を引き出し、各素子の端子へ接続する場合、図5Bのように幹線と端子は、個々に支線で配線する事が、電流容量や特性面から望ましい。しかし、従来の自動配線アルゴリズムでは配線の評価関数が配線長や混雑度をパラメータとして利用しており、図5Aの様に一筆書きで配線を行ってしまう。

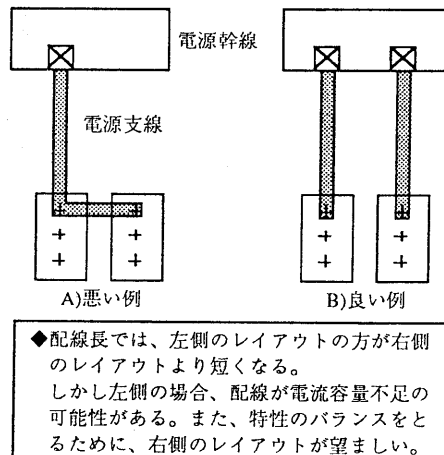


図5.電源支線配線例

本自動配線では配線経路探索時に、電源幹線方向の探索を優先する事で、所望の配線を実現している。

### 5. 2. PAD周辺用特殊処理

PAD周辺の配線は、配線方向や配線幅などに特別なルールがあるが、セル毎にルールは規格化されており自動配線が可能である。本自動配線で実現しているのは、図6のように保護抵抗および保護ダイオードを使用する場合のPADまでの配線幅と、PADからチップ内部への配線引き出し方向、および配線引き出し時の配線幅のテーパリングである。[図6]

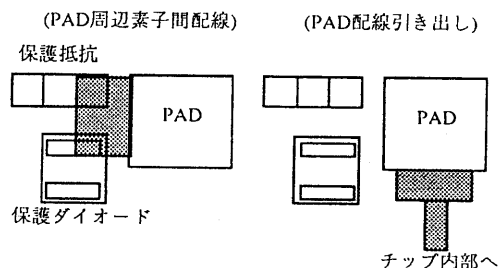


図6.PAD周辺配線例

### 5. 3. エリアコンタクト

アナログ回路のレイアウトは、幅広配線や容量のような幅広端子を持つセルが多い。従来自動配線では、配線や端子のセンター間を接続しているが、アナログレイアウトにおいて同様の処理をおこなうと、図7Aのように面積的な損失が大きくなる。

そこで本システムでは、容量などの幅広端子や幅広配線については、領域を配線対象として認識し、領域として接続した所で配線を止める事で、図7Bのように面積効率の良い配線を実現している。

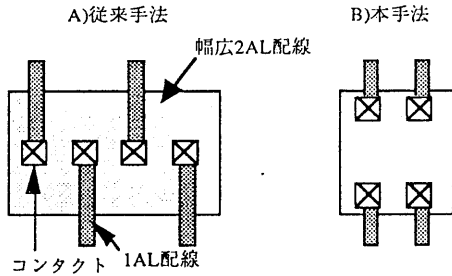


図7.エリアコンタクト例

### 5. 4. ストレッチャブルコンタクト

我々のアナログマスタースライスでは、レイアウトにおいて図8のように、

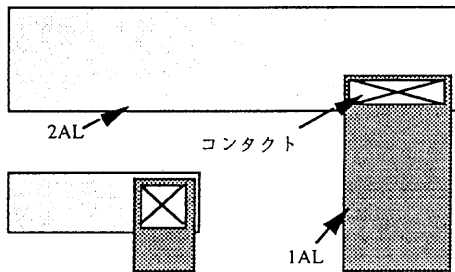


図8.ストレッチャブルコンタクト例

コンタクトサイズを配線幅に応じて可変にするストレッチャブルコンタクトを採用している。

ストレッチャブルコンタクトのサイズは、同一ネットでの配線層乗り換えが生じた時に、相互の配線層の重なり領域が、基本コンタクトのサイズより大きい場合に、重なり領域内に配置できるように決定している。

また、並列抵抗が電源やグラウンドと接続する場合に、我々の下地では並列抵抗が隣接配置されているために、個々に支線を発生させる事が困難である。このようなケースでは図9のように、幅の広い支線によって並列接続抵抗を一括配線し、幹線との間に大きなストレッチャブルコンタクトを1個発生させることによって、接続している。

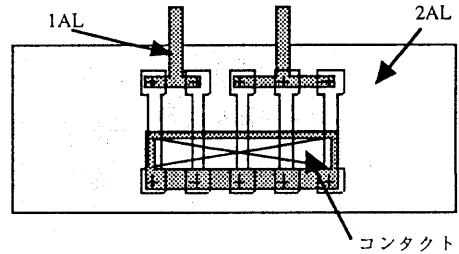


図9.並列抵抗配線処理例

### 5. 5. 抵抗素子の最適整列処理

アナログ回路において抵抗素子は、パラメーターが多く、マッピング素子の構成も基本素子の並列接続や直列接続、および直列と並列を組み合わせた物まで様々である。また、回路上で抵抗素子は端子極性が無いが、データベース上ではネット情報作成時に、設計者の意図とは関係なく端子の接続先を確定している。よって、端子の接続先まで考慮して多数の抵抗素子を配置する事は、設計者にとって、工数的にも精神的にも負担が大

きい。

本自動配線では図10のように、配線時に配線長をキーにして抵抗素子のミラーや隣接配置された抵抗素子間の交換を行う抵抗素子の最適整列化を行い、人手配置におけるユーザーの負担軽減を実現している。

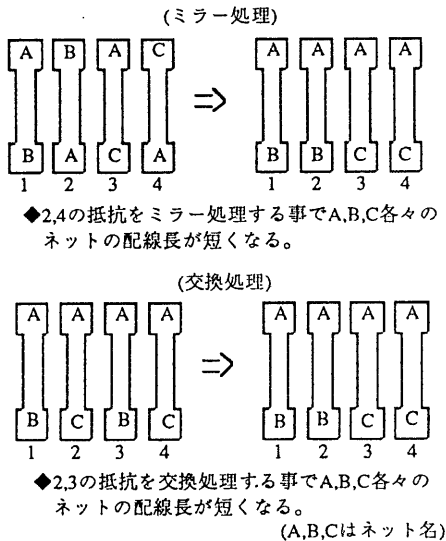


図10.抵抗素子の最適整列処理例

## 6. 適用結果

本システムをRGBコントロールLSI(2,200レイアウト素子、2電源)のレイアウト設計に適用した。図11に設計工数を従来手法(人手設計)と比較したものを示し、図12にレイアウトしたLSIチップを示す。

実行環境は、Sun社のワークステーション(SparcStation10)で、プロセスは2層アルミ配線のバイポーラプロセス、下地は1万素子である。プリ配線としては、2系統の電源用に幹線周辺に手を入れた以外は、全て自動で配線を行っている。

自動配線は、接続情報の読み出しや、

配線結果をレイアウトデータベースへ反映させる等のインターフェース処理も含めて、30分で処理を終えている。未配線は817ネットに対して19本であった。これらの未配線については、ネットリストエディット機能を使って接続した。

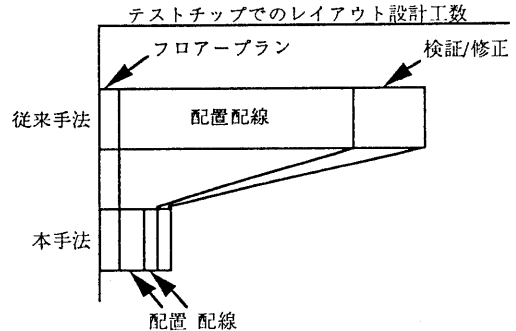


図11.レイアウト設計工数の比較

設計工数は、従来設計手法に比べて約1/3に削減されている。図11からも明らかなように、配置配線作業工数が大幅に削減されているが、配置作業は本システムにおいても人手で行われており、従来手法と同等の作業時間を費やしている。よって、自動配線による配線作業の高速化が工数削減の主要因であることが分かる。また、検証/修正作業工数が減

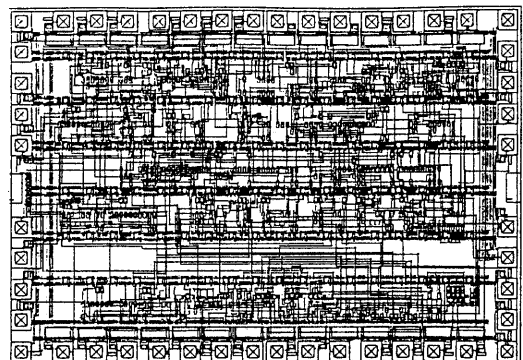


図12.レイアウト結果

少しているが、これは自動化による人手修正作業の減少に伴い、レイアウトエラーが減った為である。

レイアウトの品質は従来手法と同等で、未配線対応以外に特に回路上の問題に起因した配線修正を必要とする箇所は見つからなかった。

本システムを他のいくつかのLSIのレイアウトに適用した結果、いずれの場合もレイアウト設計工数が、従来手法に比べて約1/3に削減される事が確認された。

## 7. まとめ

今回、バイポーラプロセスを使ったアナログマスタースライス用の自動レイアウトシステムを構築するにあたり、アナログ制約等特殊な配線（交差禁止、対象配線等）などは、予め人手で配線可能にし、そのレイアウトを保ったまま、一般配線の自動配線処理を行うという人手と自動の混在を許したレイアウトシステムを開発する事で、複雑なアナログ制約を考慮しながらアナログレイアウトの自動化を実現した。

設計工数の大部分を占める配線処理部分を自動化する際も、電源配線への一方支線接続、PAD周辺での特殊処理、エリアコンタクト、ストレッチャブルコンタクト等の処理や、抵抗素子の最適整列処理など、随所にデジタル系の自動配線とは異なる機能を開発し、アナログレイアウト自動配線処理を可能にした。

さらに、回路素子とレイアウト素子の対応をライブラリー化（デバイスマッピング）、回路の素子に対応したレイアウト素子の配置（スキマティックドリブンレイアウト）、回路接続情報を保った配線修正（ネットリストエディット）等、回路とレイアウトの対応を保ったまま

のレイアウト処理を可能にし、システム全体での回路とレイアウトの整合性を保つ事により、人手介入によるエラーの混入を防ぎ、配線処理の自動化とともに設計工数の大幅な削減を実現している。

以上により、アナログマスタースライスLSIにおいて、人手と同等品質のレイアウトを従来の設計工数の1/3で実現できた。

今後の課題は、

- ・自動設計向き下地パターンの開発
- ・素子配置用の配線混雑度見積り機能など人手設計支援機能の開発

であり、これらの実現によって、より少ない工数にだれでも設計が出来るようなシステムが可能になると思われる。また、今回開発した自動配線機能は、アナログカスタムLSIへも適用が可能であり、アナログカスタムLSIにおいても、レイアウト設計工数の削減を図っていきたい。

## 8. 参考文献

- [1]小島他、”線分展開法の改良とその評価”、設計自動化、Vol.48, No.6, pp.18(1989)