

階層構造を利用した デザインルールチェックシステム

小林裕一 小池三博 高島誠
(株東芝 集積回路事業部)

1. はじめに

LSI, VLSI へと集積される回路規模の増大に伴い、マスクパターンデータに対するCAD処理時間がレイアウト開発期間のかなりの部分を占めるようになり、その高速化は急務である。

一般に、LSIにおけるマスクパターンに対するCAD処理は、

① 設計規則検査(Design Rule Check)
LSI製造プロセスに依存する幾何学的規則(幅・間隔etc)の検査。

② マスクパターン補正
ウェハー仕上り寸法を最適化するために、あらかじめ、設計時のパターンデータを補正する。

③ 回路抽出と各種の電氣的検査
パターンデータより電氣的回路を抽出し、これをもとに回路接続や電気パラメータの妥当性などを検査する。

に大別できる。

これらのCAD処理は、レイアウト設計時に作成された階層構造をすべて展開したデータに対して、一括処理するのが一般的である⁽¹⁾しかし、この手法では、処理対象データが多く、処理時間が長くなる。大規模化の著しい現在のLSIでは、この手法はすでに限界にきていると言える。


一方、階層構造を利用して、セル単位に前述の処理を行う手法⁽²⁾⁽³⁾も各種発表されているが、これらはほとんど対話形式の設計を前提としていること、レイアウト構成上の制限があることなどから、現在のレイアウト設計手法にそのままではなじまず、あまり実用と

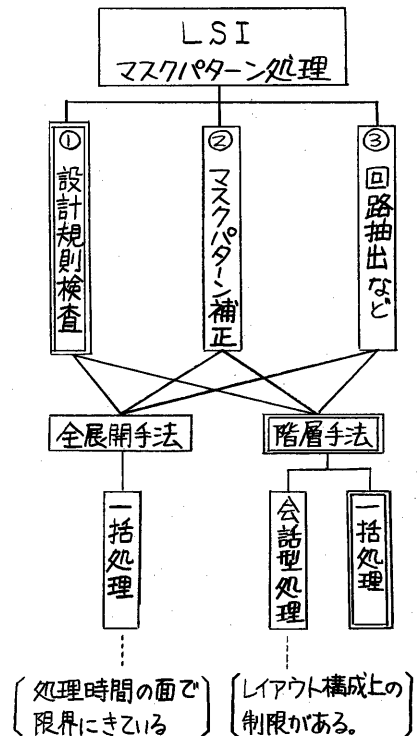
なっていない。

現在、我々は階層構造を利用することによるマスクパターン処理の高速化並びにその実用化に取り組んでいる。

本報告は、階層構造を利用した、設計規則検査を行うシステムに関するものである(図1)。

当システムでは、レイアウト構成上に制限を設けることなく、一括処理を行い、設計規則検査時間の短縮を実現している。また当システムは、ルーチン的使用を開始しており、その効果を上げ始めている。

図1. LSIのマスクパターン処理
当システムでは  を採用・実現している。



2. レイアウトデータの階層構造と階層的処理手法の特徴

2.1 データ構成

通常、CALMA等のグラフィックシステムで作成されるLSIのレイアウトデータは、階層構造を持つセルの集合である。

各セルは、①パターン情報
②セル参照情報
で構成されている。

1つのLSIチップに対応するセルをルートセルと呼び、このルートセルを頂点とするセル参照情報によるツリー構造が、レイアウトデータの階層構造である(図2)。

また、セル間の参照・被参照関係において、参照する側のセルを親セル、参照される側のセルを子セルと呼んでいる。

2.2 全展開手法と階層処理手法

従来から行われている全展開手法によるパターン処理では、セル参照情報をすべてルートセルへ展開し、これに対して各種の処理を行っている。

一方、階層処理手法とは、レイアウトデータの階層構造を何らかの方法で利用することにより、その処理の対象とするデータを、

- ①各セル固有のパターン
- ②親セルのパターンと、子セルの接続部分

に限定し、パターン処理の高速化をはかる手法である(図3)。

2.3 階層処理と処理時間

階層処理は、前述の通り、処理時間の短縮を目的としているが、その効果

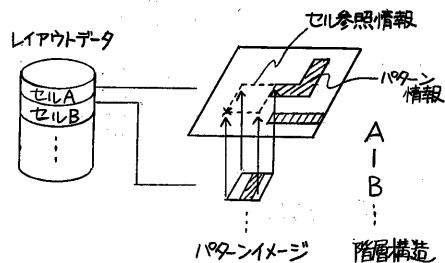
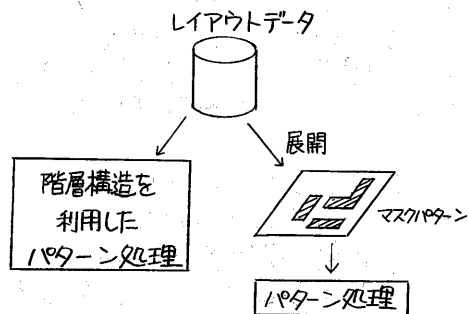


図2. データ構成



<階層処理手法>

<全展開手法>

図3. 全展開手法と階層処理手法

は、処理対象となるLSIのレイアウト構造に依存する。

階層処理の観点に立ったとき、レイアウト構造の特徴を示すものとして、Regularity^[3]と重複度が考えられる。

Regularityとは、階層処理と全展開処理の処理対象パターン数の比率を示すもので、

$$\text{Regularity} = \frac{\text{全展開パターン数}}{A \times \text{各セルのパターン数}}$$

A: ルートセルの階層構造に属するセルの集合

で表わされる。

次に、それぞれのセルのパターン存在領域の外接矩形を、そのセルの外形とすると定義する。

このとき、親セルのパターンと子セル外形の重なり(図4(a))、子セル外形どうしの重なり(図4(b))を、Overlapと呼んでいる。

重複度は、Overlapの全体に占める割

合である。

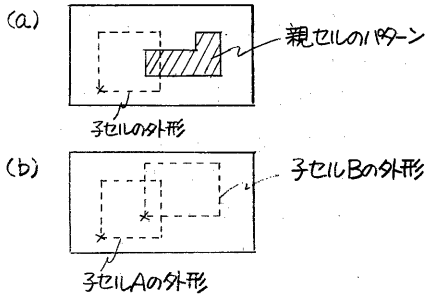


図4. overlap

Regularityが高く、重複度の少ないレイアウト構成のLSIに対しては、階層処理による大幅な処理時間の短縮が期待できる。

例えば、overlapを自由に許す場合、階層処理では、その処理対象とするパターンが多くなり処理時間も長くなる(図5, A点)。しかし、overlapをまったく許さなければ、各セル固有のパターンと、子セルの一部パターンを対象として処理すればよいので、処理時間は短くなる(図5, C点)。また、これらの中間的な制限(指定値以内のoverlapを許すetc)も考えられ、この場合、処理時間はAとCの間となる(図5, B点)。

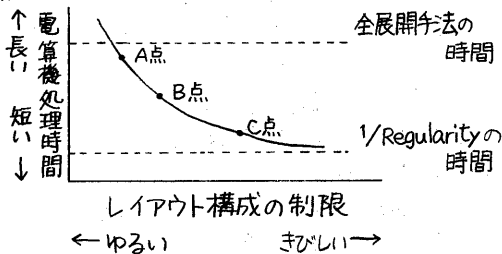


図5. レイアウト構成上の制限と処理時間

上図で示すように、レイアウト構成上の制限をきびしくすれば、階層処理の処理時間は、 $1/\text{Regularity}$ に近づくことが期待できる。

ところが、このような制限を設けると、レイアウト設計者への負担が増え

ることのほか、チップサイズの拡大につながる危険もある。このためレイアウト用CAD(配置配線etc)ツールのサポートにより、開発期間の短縮をねらったゲートレイヤスタンダードセル方式の製品ではレイアウト制限をきびしくできるが、メモリ、マイコン等の標準品では、レイアウト制限はゆるい方がよい。

従って、パターン処理用CADとしては、どのような制限の下で設計された製品も扱え、しかも、それぞれの制限の程度に応じて効果の得るCADツールが望ましい。

3. 階層構造を利用したデザインルールチェックシステムの概要

3.1 特徴

当システムは、デザインルールチェックに要する電算機処理時間の削減と現在のレイアウト設計手法への適応を目標としており、以下の特徴を持つ。

- ① 大型計算機上での一括処理が行える。
- ② 処理時間が短い。
- ③ レイアウト構成上の制限が一切ない。

通常の階層処理システムでは、overlapは許されないが、当システムでは特別な制限はない。

- ④ 検出エラー数が少ない。

従来はセル内で発生した違反は、そのセルの参照数だけ違反が表示されるが、当システムでは、原則的には一ヶ所である。

設計者にとって検出しやすい。

- ⑤ 従来のデザインルールチェック用コマンド系列をそのまま使用できる。

次節で示すように、当システムでは、従来手法での全展開処理を

擬似展開処理に置きかえ、デザインルールチェッカは従来通りのものを使用する。従って、デザインルールチェック用コマンド系列は一切の変更なしに使用できる。

3.2 基本思想

従来のデザインルールチェックは、階層構造を持つレイアウトデータ(図6 A)を、すべて展開し、これに対してデザインルールチェッカで検証を行っていた。

当システムでは、この全展開にあたる部分を“擬似展開”に置き換えることにより、階層的なデザインルールチェックを実現している。“擬似展開”は機能的には“ふるい”の役目をはたしている。つまり、はじめ“ふるい”の中には、全マスクパターンデータが入っているが、これを階層構造を利用した処理により、冗長データは“ふるい”に残し、検査する必要のあるデータのみを通過させる。通過したデータ(図6 B)は、階層構造を保持しておらず、このデータに対して従来と同じデザインルールチェッカにかけられる。

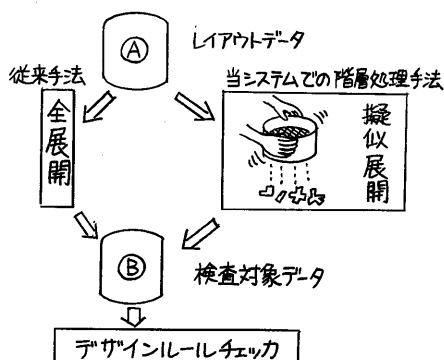


図6. 基本思想

3.3 システム構成

当システムは、既存のデザインルー

ルチェッカ(パターン処理の総合システムEMAPTM)を使用すると、その前処理としての階層処理用ソフトウェアで構成される(図7)。

また、階層処理用ソフトウェアは、

① 階層処理用 utility プログラム群

1. 階層構造の分析
2. 階層構造の整理
3. セル単位の図形分割

② 階層DRC用プログラム群

・擬似展開

で構成される。

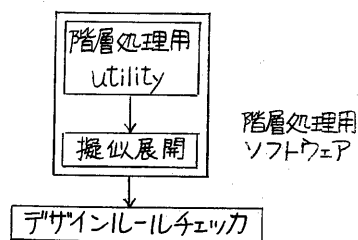


図7. システム構成

①-1. 階層構造の分析

Regularity、重複度、アレイ使用率、各セルごとの参照セル数・被参照回数、階層構造の表示など、各種情報を出力する。

階層処理の有効性の予測データとする他、レイアウト構造に関するドキュメント等に用いる。

①-2. 階層構造の整理

擬似展開処理の高速化のため、階層構造の整理を行う。

次の機能があり、任意に指定できる。

・特定セルの展開

シンボリック設計などで用いるトランジスタセル、コンタクトセル等小さなセルは、擬似展開に先だち、あらかじめ展開しておく。

・アレイデータの処理

アレイデータは普通、アレイを構成する全セルを検査する必要はなく、他との接続部分、すなわち

周囲のセルのみ、を検査すればよいことが多い。表1で示すようにユーザー指定に応じてアレイデータの前処理を行う。

・参照数が1つのセルの展開

参照数が1つのセルに、階層処理は不要である。これらのセルは展開する方が、階層構造が簡単になる。

表1. アレイデータ処理

| | モード | 処理内容 |
|------------|-----|----------------|
| 一次元 アレイ | A | 個々のセルにする |
| | B | 削除する |
| | C | 前後2セルを残し他を削除する |
| 二次元 アレイ | A | 個々のセルにする |
| | B | 削除する |
| | C | 周囲一列を残し他を削除する |
| | D | 周囲二列を残し他を削除する |

①-3. セル単位の図形分割

検査対象データの量を減らすため、擬似展開処理に先立ち、セル外形から指定値分内部の位置でパターンを分割を行う(図8(a))。図8(b)に示すように、セル単位の図形分割後、擬似展開を行うと通過するパターンの座標点数が減り、これにより検査対象のデータ量も減少する。

図11(c)にセル単位の図形分割の後、擬似展開を行ったLSI-Dの全体図を示す。

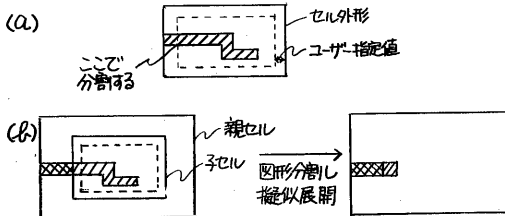


図8. 図形分割

②. 擬似展開

当システムのメイン処理部分で、冗長データを残し、検査する必要のあるデータのみ通過させる(図10,11参照)。

オプションとして、指定層データをすべて通過させることもできる。これは、たとえばCMOSのPwellやNM

OSのD-typeのように、トランジスタの種類分けを行う層データに対して適用している。

3.4 擬似展開のアルゴリズム

overlapを禁止していれば、セル周辺上のパターンのみ通過させればよいが当システムでは、前述の通り、これらの禁止項目がなくても対処できるように考慮している。

基本的な考え方は次の通りである。

図9参照。

i) 複数個配置されているセル参照のうち、1つの全パターンを通過させる。これは、セル内部を検査するために必要である。

ii) これ以外のセル参照に対しては、他セルのパターンと設計規則違反を起こす可能性のある図形のみを通過を通過させる。

具体的には、対象となるセル参照の外形を、近傍値(後述)の最大値で太め、これと重複のある他セルのパターンを取り出す。さらに、この図形を近傍値分太め、これと対象セル内の個々のパターンを比較し、重複があれば、そのパターンを通過させる。

上記の処理のため次のデータを入力する。

基準層 L_i ; セルの近傍、または重なりを持つ他セルのパターン層。

図9 部分。

グループ層 L_{ij} ; 基準層と何らかの設計基準のある層。

近傍値 D_{ij} ; 基準層 L_i が、グループ層 L_{ij} に対して、設計基準違反を起こす可能性のある危険範囲。

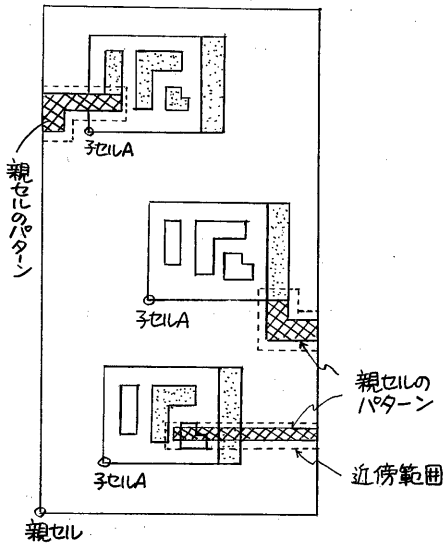
図9 部分。


N-ch Si Gateの一例を表2に示す。

表2. 入力データ

| Li | 拡散 | D-type | poly | コンタクト | Metal |
|--------|----|--------|------|-------|-------|
| 拡散 | 3λ | 3λ | 3λ | 4λ | — |
| D-type | 3λ | — | 3λ | — | — |
| poly | 3λ | 3λ | 3λ | 4λ | — |
| コンタクト | 4λ | — | 4λ | 5λ | 3λ |
| Metal | — | — | — | 3λ | 2λ |

図9. 擬似展開説明図



上図では、子セルAが3ヶ所で参照されている。上側の参照セルは全パターンが通過する。中央と下側の参照セルでは、外部図形の近傍範囲内にあるパターン()だけが通過する。

4. 階層構造を利用したデザインルールチェックの実行結果と考察

4.1 電算機処理時間

実際のLSIでの実験結果を表3に示す。いずれのLSIも、既に開発中で、レイアウト構成上の制限はなく、overlapが比較的多いと思われる。このため、通過図形数はRegularityに比べて多くなっている。

擬似展開処理時間は、デザインルールチェックに要する時間の $\frac{1}{2}$ ~ $\frac{1}{3}$ 程度

で当処の予想よりも多い。

従来手法(全展開)との総合的な処理時間比は、60~70%であるが、設計手法が改善されれば、さらに減少するものと思われる。

表3. 実LSIでの実行結果

| | A | B | C | D | |
|--------------|--------------------|--------|---------|---------|--------|
| プロセス | N-MOS | C-MOS | C-MOS | C-MOS | |
| レギュラリティ | 5.2 | 5.0 | 5.9 | 3.3 | |
| 従来手法 | 対象図形数 | 62,700 | 303,000 | 118,000 | 39,400 |
| | DRC CPU TIME | 1.40H | 4.15H | 7.60H | 0.98H |
| 階層手法 | 対象図形数 | 30,500 | 121,000 | 60,700 | 23,100 |
| | Filtering CPU TIME | 0.23H | 1.32H | 0.76H | 0.13H |
| | DRC CPU TIME | 0.69H | 1.49H | 4.1H | 0.48H |
| 従来と階層の図形数比 | 0.48 | 0.40 | 0.51 | 0.58 | |
| 従来と階層のCPU時間比 | 0.65 | 0.67 | 0.63 | 0.62 | |

4.2 検出エラー数

表4は、LSI-Cでの従来手法(全展開)と階層手法で、それぞれ検出された違反図形数である。

ほとんどの項目でエラー件数が減少しているが、項目NO.1では階層手法による擬似エラーが発生し、従来手法よりも件数としては多くなっている。

なお、階層手法でも、本来の設計規則違反は、すべて検出されていることを、違反図形のプロット図で確認した。

表4. 検出エラー数

| NO. | 従来手法のエラー数 | 階層手法のエラー数 | 判定 |
|-----|-----------|-----------|----|
| 1 | 17 | 20 | △ |
| 2 | 10 | 10 | ○ |
| 3 | 13 | 7 | ○ |
| 4 | 11 | 5 | △ |
| 5 | 7 | 7 | ○ |
| 6 | 0 | 0 | ○ |
| 7 | 242 | 126 | ○ |
| 8 | 7 | 7 | ○ |
| 9 | 19 | 19 | ○ |
| 10 | 81 | 7 | ○ |

- 階層手法ですべての違反を検出かつ擬似エラーなし
- △ 階層手法ですべての違反を検出しかし擬似エラーあり

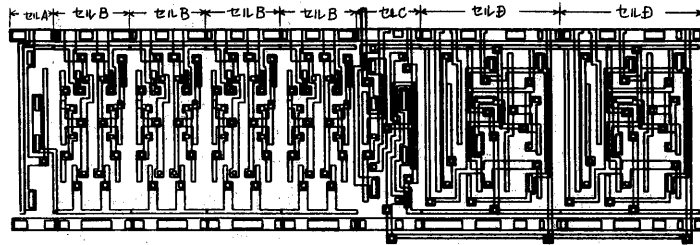


図10(a) 全展開したマスクパターン図(マクロセル)

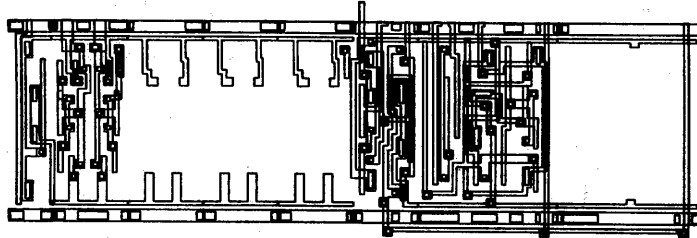


図10(b) 擬似展開後のマスクパターン図(マクロセル)

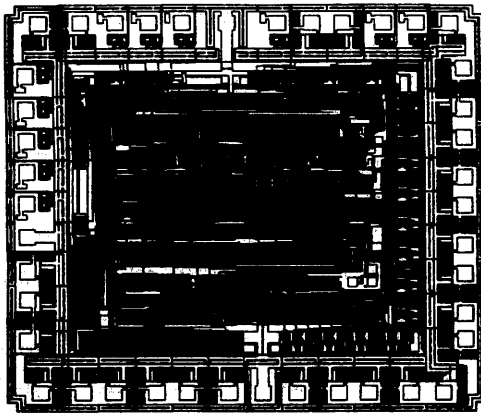


図11(a) 全展開したLSI-Dの全体図

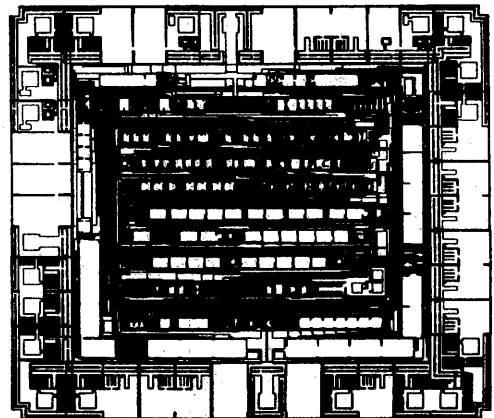


図11(c) セル単位の図形分割・
擬似展開後のLSI-Dの全体図

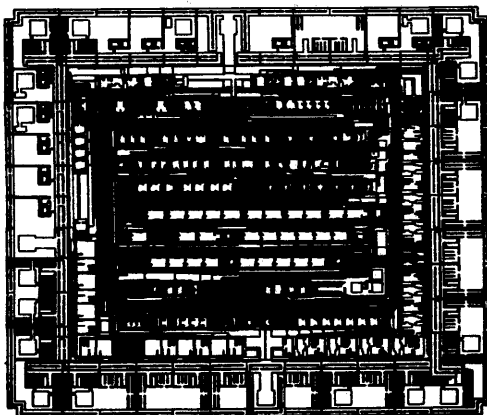
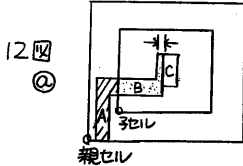


図11(b) 擬似展開後のLSI-Dの全体図

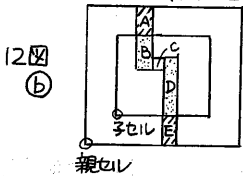
4.3 擬似エラー

次のような擬似エラーが発生した。

- ① 不要な分割線による擬似エラー
 下図で示すように不要な箇所分割線が入っていると図形Cは通過しないため、B図形は幅違反となる。

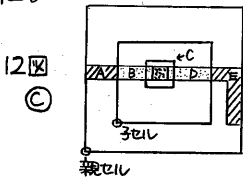


- ② 同様にC図形が通過しないと、B図形とD図形の間隔違反となる。



- ③ B図形(拡散)とD図形(Poly)は通過するがC図形(burried contact)が通過しないと、不要な拡散とPolyの重複部として違反となる。

N-chで、この種の擬似エラーが多発した。



5. まとめ

当手法による階層的なデザインルールチェックは、レイアウト構成上の制限がないため設計済のLSIにすぐ適用できることや、従来の処理フローやデザインルールチェックのコマンドシーケンスをそのまま使えるという点でも、非常に実用化しやすい。

当システムにより、現状でも電算機処理時間を60~70%に短縮できた。レイアウト構成上の制限を設ければさらに減少し、1/Regularityに近づいて行く

ことが期待できる。

階層手法により発生する擬似エラーは、特定層の全データを通過させたりセル内で図形の合成等、行ってから擬似展開すれば避けられるが、処理時間が増加する。今後検討すべき問題である。

最後に、当システムの開発に協力していただいた東芝マイコンエンジニアリング(株)、東芝エンジニアリング(株)の関係諸氏に感謝します。

参考文献

1. 千葉他「LSIマスクアートワーク解析システムとその応用」1980年12月電子通信学会、電子装置技術研究会 7-3
2. S.N. Stevens, "A Hierarchical Approach for Mask Verification" Proc. of ISCAS, pp.702-705
3. W.J. McCalla, "Symbolic Representation And Incremental DRC for Interactive Layout", Proc. of ISCAS pp.710-715 1981
4. T. Whitney, "A Hierarchical Design-rule Checking Algorithm", LAMBDA pp.40-43, 1st Quarter 1981
5. T.J. Wagner, "Hierarchical Layout Verification" Proc. of 21st DA Conference, pp.484-489 1984