

MOS-LSIレイアウト設計システム LILAC-3の開発

石賀忠勝*

小澤時典*

佐藤昭治*

酒見淳也*

堀野 寛*

松崎常正**

(日立製作所)

* 中央研究所

** 半導体事業部)

1. まえがき

最近のMOS-LSIのレイアウト設計では、LSIの集積度が増大する一方で高集積密度とターンアラウンド時間短縮の要請が強まってきている。標準セル方式によってレイアウト設計を行なうようになってから、この設計工程をCAD化する試みが多数行なわれてきた。筆者らが1972年に発表したLILAC-1 (Logic-Input Layout Automation with Cellular Approach) システム¹⁾もこのようなCADシステムのひとつである。しかしながら、LSIの設計内容は、単に集積度増大という変化にとどまらず、ROM (Read Only Memory)、PLA (Programmable Logic Array)、SR (Shift Register) などの論理ブロックを多く含むようになってきて、セル方式のみを対象とするだけでは済まなくなってきた。このような事態に対応するために新たにLILAC-2の開発を行なった。²⁾ このシステムでは、対象とするLSIの規模を大きくし、モデルの自由度も増し、処理上の多くの新しい手法を使っていた。

LILAC-3はこれらのシステムの経験と評価を踏まえて、次の目標のもとに開発したシステムである。その目標は、

- (1) アルゴリズムを改良して完全な自動化が行なえ、また高集積密度 (小チップサイズ) を得ること、
 - (2) 自由なレイアウトの修正指示によって、各品種ごとに異なる条件にきめ細かく追従して設計品質をあげられること、
- の2点が主たるものである。

2. システムの概要

2.1 LSIのレイアウト・モデル

CADシステムにおけるLSIのモデル化は、システムの特性を決定する基本的な要素である。すなわちシステムの外部に向っては、システムの適用範囲と自由度を決定するのであり、できるだけ一般的 (あるいは汎用的) であることが望ましい。システム内部では、解くべき問題の内容を決定し、またモデルの記述に必要なファイル内容、構造を左右するのである。このとき処理すべき問題は、計算機の現実的な処理能力 (メモリ量と処理時間) に見合う形で定式化される必要がある。最近の (あるいは今後の) LSIはきわめて大規模であるから全体的 (トータル) な最適化が難しく、適切な設計過程の区分が行なえて、各過程での最適化が図れるようなモデルとそれに基づく定式化が要求される。

セル方式の最も基本的なレイアウト・モデルは図1に示すようなものである。このモデルでは配線領域を挟んで向い合う素子列の組を1ブロックとし、数個のブロックが隣接して横に並ぶ形になっている。これによって問題を1ブロックごとの処理に区分けると合せて (共通線=ブロック間配線の処理も1ブロックの配線と見なすことができる)、設計過程を論理分割、共通線割当て、配置、配線

と続く一連の処理に整理することを可能にしている。

先に述べたPLA (ROM, SRも含めての統称とする)は,

- (1) セルに比してサイズが大きい,
- (2) 入出力端子数が多い,
- (3) PLA内部のパターンはそこで使う単位素子の種類によって多様であるが, パターンそのものは規則的なくり返し配列である,

などの性質をもっている。PLAをセルと混在する形でモデルに取りこむために, 周囲に端子群を備えた矩形でPLAを表現することにした。そしてそれが素子列と組合わさる形式を3通りに整理してタイプ1~3のPLAとした。したがってこのモデルでは, PLAは外に向けた形状はもっているが, その内部は別に設計することになっている。

図2はPLAを含むLSIモデルである。3種のタイプのPLAは次のような性質をもっている。

- (1) タイプ1: 2列以上の素子列にまたがる大きなPLAであり, このPLAの位置は入力データで指定する。
- (2) タイプ2: 比較的小さいPLAに当り, 素子列の一部に組込む形状になっている。この位置は指定してもしなくてもよい。
- (3) タイプ3: 1列の素子列全体とおきかわる形のPLAで,

シフト・レジスタを代表したり, 素子列の削除に使ったりする。

PLAとは異なるが, 背中合せにある素子列の間隔をあけてそこにたとえば細長くシフト・レジスタをおく余地を残すことがこのモデルで可能になっている。

設計する範囲は, LSIの周辺部(ボンディング・パッド, それとの入出力接続線など)を除いた, いわば活性領域と呼ぶ範囲である。周辺部とPLAの中味とは人手の設計に残されている。

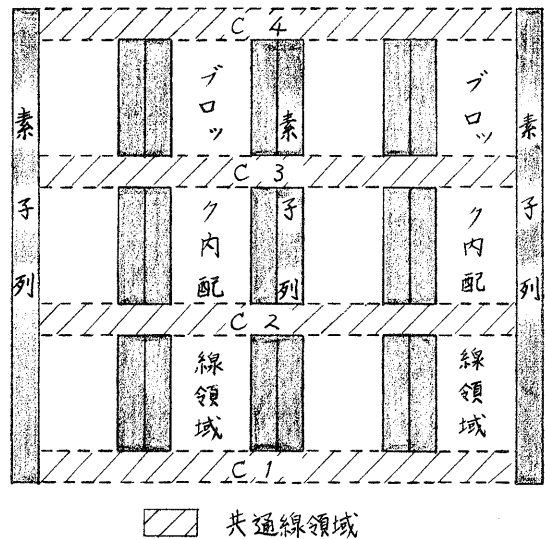


図1. セル方式によるLSIモデル

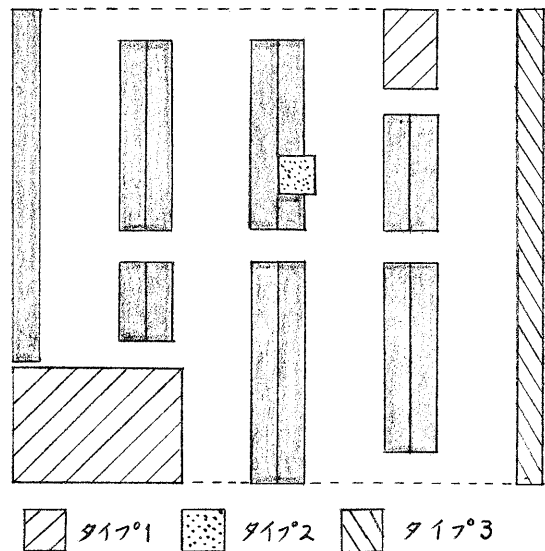


図2. PLAを含むLSIモデル

2.2 システムの構成

LILACシステムは論理結線関係の入を与えると、自動的にレイアウト・パターンを出力するシステムである。純然たるセル方式のときは論理ゲートに対応するセルのライブラリを使って自動設計ができる。LILAC-3ではそのほかにPLAに関する入力データが必要である。

論理の入力後の処理は(1)論理分割、(2)共通線割当て、(3)配置、(4)配線、の順序で行なう。図3はこれを図示したものである。

各処理ごとに必要な入力データをすべて用意しておけば、一回のランで最終出力、すなわちプロッタによって描かれたレイアウト・パターン図が得られる。しかし各過程の結果をリストに出してこれによって設計の良否を判断しながら、さらに先に進むか、それ以前の処理にさかのぼって別のパラメータで再ランさせるか決定することができる。

図3に示す各種の修正処理も同じような形式で設計者の判断に基づく介入を行なうものである。ただし修正処理では、前回の処理結果のファイルを生かして、これに基づいて変更(修正)を行なう点が異なっている。

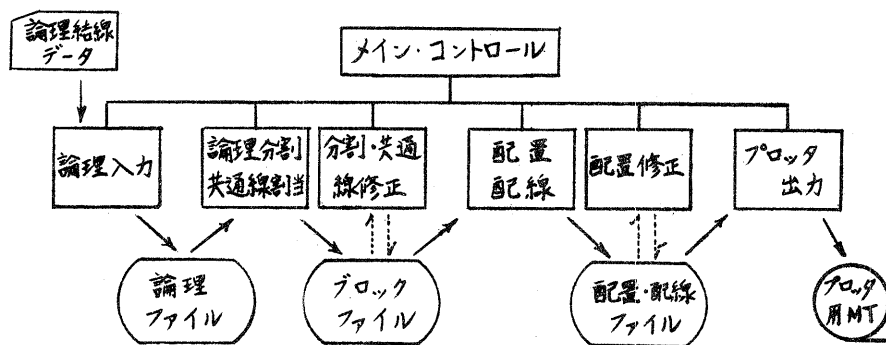


図3. LILAC-3 システムの構成

3. アルゴリズム

ここでは各処理過程ごとの解くべき問題を述べ、本システムで採用しているアルゴリズムについて概略を説明する。

3.1 論理分割

論理分割の目的は論理図全体を数個のブロックに分けることである。この時タイプ1, 3のように大きいPLAは置くべき位置の指定があるので、分割の対象はランダム・ロジック部分とタイプ2 PLAになる。分割する際の目標は、各ブロックが同サイズであって、同時にブロック間の配線(すなわち共通線)の本数を最小にすることである。ブロック・サイズの均等化は共通線が配線された時に占めるサイズを予測に入れて行なうもので、これによってデッド・スペースの生ずることを避けるのである。共通線本数の最小は一般的な分割の目標と言えるが、本システムで採用しているレイアウト・モデルと幹線支線配線方式³⁾によれば、共通線の増加は幹線本数の増加を直接に意味し、これはチップ・サイズ増大につ

なものである。

本システムではクラスタ入替え法⁴⁾によって論理分割を行なう。クラスタとはまとまりのよい素子グループを指している。分割すべき論理図の全体を覆うようにほぼ均一なサイズのクラスタを作成してから、次に入替え法と呼ぶ繰返し修正により分割を改良して良い結果を求める手法である。

クラスタの作成には素子グループのまとまりの良さを評価する関数が必要である。そのような関数として

$$\eta = 1 - \frac{\text{素子グループ外へ向う配線本数}}{\text{素子グループ内総端子数}}$$

を採用している。

この η は、 $0 \leq \eta \leq 1$ の値をとり、1に近いほどまとまりはよいと判定する。図4の例では、 $\eta = 1 - 5/14 = 0.643$ である。

入替え法では、初期(仮)分割から始める(2分割とする)。次いで互いの組の中から相手の組に移ることがその間の共通線を減らすようなクラスタを入替え候補として取出す。この候補(複数)の中から適当な対を選んで入替えを行なわせる。入替えが進んで候補が尽きるとそれまでの入替えで各クラスタの条件がかわっていて、新しい候補クラスタの可能性が出る。そこでこの状態を新しい仮分割として次のサイクルに入る。この繰返しのうちに候補があがらなくなると分割が終る。多分割は2分割を重ねることで可能である。

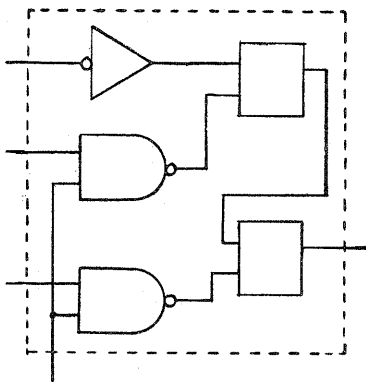


図4. クラスタの例

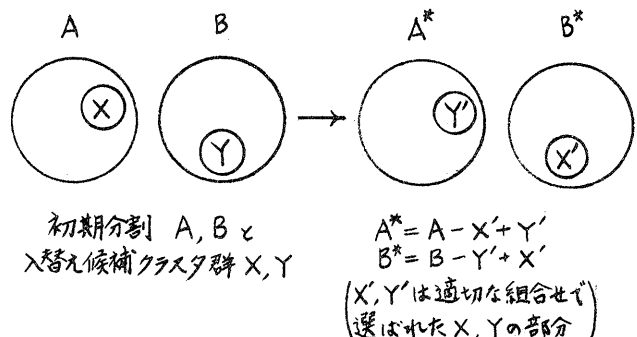


図5. クラスタ入替え過程のトサイクル

3.2 共通線割当て

この過程で共通線の領域の形状(素子列, PLAとの組合せ形状)を決定し、次いで各共通線の通るべき領域への割当てを行なう。共通線の割当ては配置と独立に最適化することはできないが、本システムでは共通線割当てを先に行なって、その結果が配置にとっての制約であるとみなしている。

共通線割当ての目的は遠まわりになる共通線の現れないように割当てを行なうことである。そのために共通線同士が各ブロック内の素子を仲介にしている関係(同一素子と共通に接続している関係)に注目して共通線をグループ化し、これに基づいて割当てを行なう。その順序は次のようになっている。⁵⁾

(1) 共通線領域の形状決定: 共通線本数, 素子列Y方向長, PLAの配置, な

どに基づいて決定する。

- (2) 外部端子位置決定：各ブロックから直接に外部（ボンディング・パッド）に向う配線を上下いずれに出すか決定する。これはブロック内部の仮配置を行なったうえで決定する。
- (3) 結合グループの作成：先に述べた共通線の関係を取り出してグループ化する。
- (4) 割当て決定：結合グループを単位に仮配置の結果に基づいて全ブロックにとって最も望ましい領域を決定する。

3.3 配置

ブロック内配線領域の最小化が配置の目標である。しかしこの目的に直接達することは難しい。そこでブロック内配線領域のX方向長（幅）の値を決定するのが幹線であることに着目し、幹線長を最小とすることでこの処理の目標にして配置を行なう。

配置は繰返し改良する方法で行なう。すなわちある配置から、各素子にとって幹線長を減らすような移動方向（最適な座標）を求めて移動を行ない、この結果を新たな配置として次の回の素子移動を行ない、これを繰返すのである。

素子*i*にとって望ましいY座標($\bar{y}_i$)は

$$\bar{y}_i = \frac{1}{m} \sum_{k=1}^m y_k \quad (\text{素子 } i \text{ に } m \text{ 個の素子が } y_1 \sim y_m \text{ の座標で接続})$$

とする。

素子配置は次の手順で行なう。

- (1) 初期条件として共通線に座標を割当て、素子すべてを1列にY方向に並べた仮配置をつくる。
 - (2) 素子移動の繰返しで安定するまで配置を最適化する。
 - (3) 素子領域形状に合せて2列に素子を分配する。
 - (4) 2列の状態を配置を繰返し改良して最適化する。必要な素子反転を行なう。
- 以上の過程の中で、ある種の素子をまとめたい時に与えられた指定に基づいて配置を決定することも含せて行なうことができる。

3.4 配線

本システムでは2層を使っての幹線支線方式で配線を行なう。³⁾ ブロック内ではY方向（縦）の線が幹線、X方向（横）の線が支線であり、共通線領域ではこの逆になる。共通線領域はブロック内配線の結果を受けて配線を行なう。

配線処理は素子列端子間の接続関係で決まる最小X方向長の中で配線経路を決定することである。結線の間には2通りの制約がある。各幹線の占めるY座標を比較し同じY座標値をもつことで幹線のグループ化が行なえる。同一グループの幹線は同じX座標をもつ経路をとりえないので、各配線領域のX方向長は、各グループに含まれる幹線数の最大値より小さくはならない。もう一方は支線（すなわち素子入出力端子）のY座標と左右の位置で規定される幹線同士の左右関係である。この関係を表わす幹線間の2項関係グラフの最も長いパスがやはりX方向長の下限になる。

結線のあいだの後者の制約は、グラフにサイクルの現れる時に障害になる。もしサイクルがなければ、グラフの示す順に経路が決定できる。サイクルの除去は

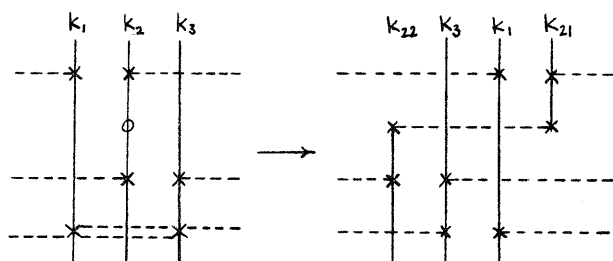
次の4通りの方法を用いて行なう。特に幹線分割Ⅱによって常にサイクル除去が確実にになり、未配線なしの100%配線を保証している。

(1) 素子の端子置換：置換可能な等価の端子を入替える。

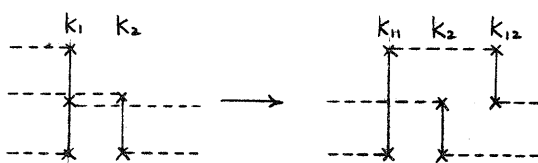
(2) 素子の反転：素子を×方向の軸でミラー反転する。

(3) 幹線分割Ⅰ：幹線を分割する。図6(a)がその例で、幹線はかわっていない。

(4) 幹線分割Ⅱ：幹線を分割すると同時に幹線を延長して別のY座標で制約を逃れる。図6(b)がその例である。



(a) 幹線分割Ⅰの例 (k₁の分割でも可能)



(b) 幹線分割Ⅱの例

図6. 幹線分割の方法

4. 修正プログラム

LEILAC-3は設計者による修正が自由に行なえる新しい機能を備えている。この修正はカード入力によって行なう。

修正プログラムの目的は次の2点にある。すなわち

(1) チップ・サイズの縮小

(2) レイアウト図の質的な内容の向上を目指している。

これらはいずれも設計者が出力レイアウト図を見て判断することで、このレイアウト図をふまえた修正、つまり前回の結果を含むファイルの一部修正を施すためにプログラムが必要なのである。

前記の目的のうちチップ・サイズの縮小はその意味が明らかであろう。後者の質的な向上とは、セルのブロック系列の区分、クリティカル・パス対策、セルの種類による隣接条件（ある組合せの場合に間隙を設ける必要がある）などの項目を指している。これらは設計図が必ず満たさなければならない基本的制約ではあるが、そのすべての一般的な形でのルール化が難しく、またLSIの品種ごとに様相が異なっているので、必要に応じて最終的には設計者が保証しなければならないのである。

本システムの全処理の中で真に設計を行なっている過程は、論理入力とプロッタ出力を除いた分割、共通線割当て、配置、配線である。このうち配線は、それに先立つ配置までの処理結果によっていわば一義的に解の決定するものであり、この段階で設計結果（レイアウト図）を左右する余地がない。したがって修正の対象から配線を除外することができ、分割、共通線、配置が修正プログラムの範囲になる。

各修正プログラムは次のような考え方で作成されている。

- (1) 論理分割：各素子ごとにブロックの帰属を与えることができる。
- (2) 共通線割当て：共通線領域形状の決定と、各共通線の領域への割当てのいずれについても自由な指定ができる。
- (3) 配置：ブロック内部での任意の配置の修正ができる。単独素子での指定、素子グループとしてのまとまった指定、いずれも可能である。ブロック相互については相対的なY座標の調整ができる。

5. LILAC-3の性能

本システムは主としてFORTRAN-IVで記述されている。約53 K ステートメントのプログラム規模である。

システムの対象であるLSIは、2000素子（論理ゲート相当）を考え、それに加えて最大200までのPLAの混在が可能である。

レイアウトCADの性能の評価は主としてチップ・サイズ、処理時間の点から行なわれる。（システムの最終的な評価は、それを利用した設計工程全体への寄与で計られるが、現在は未だその時期でない。）

チップ・サイズの比較の対象は人手設計の結果である。開発直後の試用の例では次の結果を得ている。すなわち自動設計、修正を行なった結果、対人手比が110%以下に達したものが12品種中10例である。これは人手より小さい結果も含んでいる。人手設計そのものが設計者による差異をかなりもっているので、この比較の妥当性はいくぶん減殺されるかも知れない。一般的には規模の小さいLSI、PLAの個数が少ないLSIで良い結果に達している。

計算時間はHITAC-8500を使ったとき、データ入力からプロッタ用MTの出力まで20～40分が平均的な値である。この値は十分実用的な値と見なすことができる。

設計結果の一例を図7に示す。

製品への適用を通じて性能を把握し、このシステムを含む設計工程を確立するのが今後の課題である。

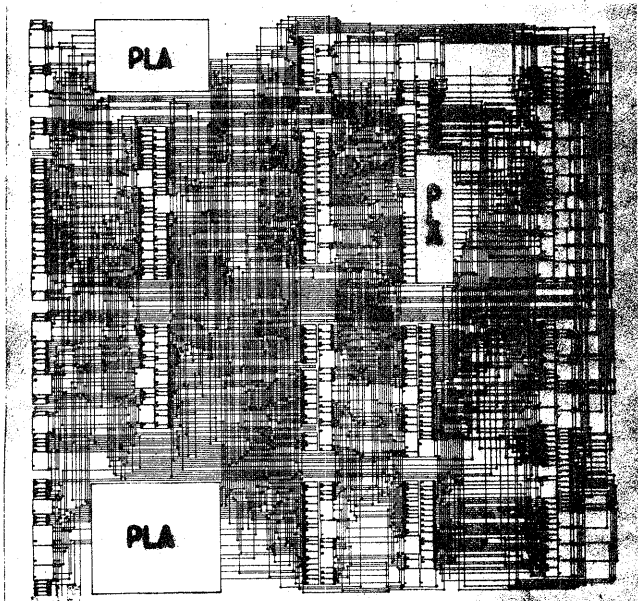


図7. LILAC-3の出力例

6. むすび

MOS-LSIのレイアウト自動化システムLILAC-3は、セルとPLAなど論理ブロックの混在するLSIを対象にしている。本システムは

- (1) 改良アルゴリズムで集積密度が向上し、
 - (2) 修正プログラムを組込んで設計者の判断に基づく自由な修正が可能である、という特長をもっている。100%配線が常に可能であることをはじめとして、各処理とも自動設計によりレイアウト図としての最終解への到達を保証している。
- 試用の結果では実際のLSIについて大部分がチップ・サイズの対人手比110%以内になっている。修正プログラムによる設計の詰めの有効性も確認されている。

今後各種品種に適用して性能を把握するとともに、ユーザ側の利用技術を高めて行く予定である。

謝辞

仕様の検討、プログラムの作成にご協力いただいた当社半導体事業部の加納主任技師、辻主任技師をはじめとする多くの方々に深く感謝する。

参考文献

- (1) T. Kozawa et al. : Block and Track Method for Automated Layout Generation of MOS-LSI Arrays ; ISSCC , p62 , (1972)
- (2) T. Kozawa et al. : Advanced LILAC - An Automated Layout Generation System for MOS/LSI , Proc. Design Automation Workshop (1974)
- (3) 浅野他4 : ビルディング・ブロック方式LSIの配線の実現可能性について , 信学会論文誌 56-A , p489 , (1973.9)
- (4) 石賀他2 : 論理分割の一方式—クラスタ入替え法— , 昭50信学会全大 , No. 409 (予定)
- (5) 酒見他3 : MOS-LSI自動レイアウト設計システムにおけるブロック間配線の領域割当ての手法 , 昭50信学会全大 , No. 404 (予定)