Vol. 26 No. 1 情報処理学会論文誌 Jan. 1985

大規模フルカスタム LSI の自動レイアウト設計方式†

寺井秀一†† 小澤時典††
坂田谷義憲††† 湯山恭史††††

集積度が数万ゲートの VLSI では人手によるレイアウト設計はほとんど不可能で計算機による自動レイアウトが必須となる．従来，自動レイアウト技術はマスタスライス（ゲートアレー）方式の LSI に対して実用化がなされてきた．しかしチップの面積効率を高めるためには個別設計型（フルカスタム）LSI が適切でありこれに対する自動レイアウトシステムが必要となってきている．本論文ではこのような大規模フルカスタム VLSI のレイアウト設計に際して発生する問題を明らかにし，これらを解決する自動設計技法およびレイアウト支援のためのいくつかのプログラムの機能，処理方式について述べる．

1. まえがき

中小型計算機や周辺装置，端末装置など，各種電子装置の小型化，低消費電力化，高信頼度化を図るには論理回路の LSI 化が不可欠である．一般に，これらの装置は生涯生産数が比較的少なく，搭載される LSI は多品種少量生産型の傾向をもつ．したがって LSI 設計コストの低減がより重要な課題となる．このような多品種 LSI の短期開発を可能とするには LSI 構成法としてマスタスライス方式の採用およびその自動レイアウト設計が効果的であり，いまでは自動配置配線プログラムはマスタスライス LSI において必須の設計ツールとなっている[1)-4)]．

一方，マスタスライスでは開発コストを優先させる結果，チップの面積効率という面ではフルカスタム LSI に比べて劣るという問題がある．たとえば中型計算機のエンジンとなる CPU などは数万ゲート規模となるが，これを数個のマスタスライスで構成するよりも1個のチップで実現するほうが LSI 化の効果は大きい．このような意味で数万ゲートクラスのフルカスタム VLSI の自動設計に対する要求が近年とみに高まってきており，自動設計による VLSI プロセッサの発表もなされている[5)]．

本論文ではこのような大規模フルカスタム VLSI のレイアウト設計に際して発生する問題を明らかにしこれらを解決する自動設計技法およびレイアウト支援のためのいくつかのプログラムの機能，処理方式について述べる．

† Automatic Layout Design Method for Full Custom VLSI by HIDEKAZU TERAI, TOKINORI KOZAWA (Central Research Laboratory, Hitachi, Ltd.), YOSHINORI SAKATAYA (Kanagawa Works, Hitachi, Ltd.) and KOUJI YUYAMA (Device Development Center, Hitachi, Ltd.).

†† 日立製作所中央研究所
††† 日立製作所神奈川工場
†††† 日立製作所デバイス開発センタ

2. レイアウト設計階層

チップ内に搭載可能なトランジスタの個数が数十万のオーダに達しつつある状況では，チップ全体を一挙にレイアウトすることは設計管理の限界を越えて設計そのものが収束しなくなるおそれが出てくる．この問題に対して通常とられる方策が，いわゆる階層設計法である[6),7)]．階層レイアウト設計では，チップの構成要素にマクロなレベルからミクロなレベルに至る包含関係をもたせ，各レベル（階層）内での設計を積み重ねながら全体のチップレイアウトを実現する．ここで採用した論理およびレイアウトの階層を図1に示す．図1でチップは複数個のブロックの部分集合に分割可能である．この分割された一つをブロックアセンブリ（あるいはたんにアセンブリ）と呼ぶことにする．階層化の概念は対象の深さ方向への分割と考えられるのに対して，ブロックアセンブリへの分割は一つの階層内での対象の分割である．このため設計ファイルの分割/併合機能が用意されている．

ブロックアセンブリ導入の狙いは次のとおりである．

1) 計算機メモリネックの解消

ブロックの配置設計はチップ全体を対象にするのがチップ構成を最適化するうえからも妥当である．しかし，一度配置が決定したブロック間の配線設計は必ずしもチップ全体を処理対象とする必要はない．逆に，配線格子数増大によるいくつかの問題が発生する．その一つは配線処理に必要な CPU メモリ量の問題である．たとえば，一辺の長さが 10 mm のチップを 3 μm ピッチで配線するとすればチップ全体での配線格子数

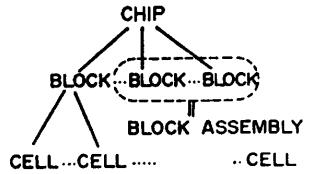


図 1 レイアウト設計階層
Fig. 1 Layout hierarchy of chip.

は 3,000×3,000 格子となる．これを Lee アルゴリズムで自動配線すると配線メッシュテーブル（計算機メモリ内部にもつ LSI 内部の配線領域と相似な 2 次元テーブルで，1 エントリ（2 バイト）が 1 配線格子に対応する）だけで 18 M バイト必要となる．これは現在の大型計算機の CPU 記憶容量を越えており，自動配線を可能とするためにはチップの細分割が必要となる．

2) 作業性と設計効率の向上

1)のような計算機メモリの制限からくる問題以外に，レイアウトの作業性と，効率を高める工夫として分割を積極的に用いる．すなわちプロット図面の巨大化抑制および複数設計者の同時並行作業による設計期間の短縮である．設計対象を分割することによってこれらが可能となる．

3. レイアウトモデル

人手によるレイアウト設計（マニュアルレイアウト設計）の場合は，製造プロセス上規定されるレイアウトルール（プロセス設計基準）を満たす範囲内できわめて柔軟なレイアウト設計が可能である．これに対し，計算機による自動レイアウトを実現するには，その手順が完全にプログラムとして記述できることが必要であり，このためにはレイアウト方式について種々

表 1 レイアウト階層と配線層の関係
Table 1 Relation between layout hierarchy and wiring layers.

	Poly Si	1st metal	2nd metal
Cell	○	○	×
Block	○	○	○
Chip	×	○	○

○：使用可，×：使用不可

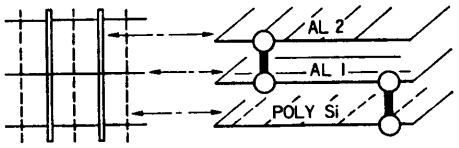


図 2 配線格子モデル
Fig. 2 Wiring layer model.

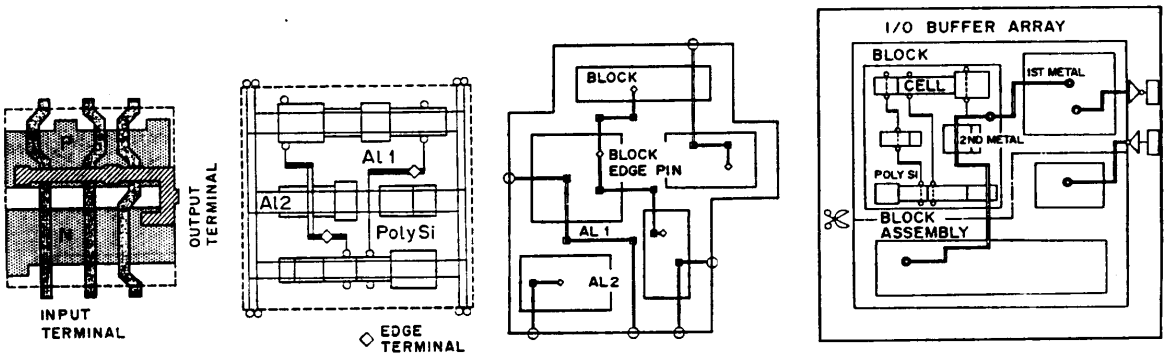
の取決めを定式化した規範的概念，すなわち「レイアウトモデル」を確立しておくことが必要となる．レイアウトモデルでは，プロセス設計基準をさらに計算機処理上から見直して規格化したレイアウトルールとそれに準拠したレイアウト形態の二つが定義される．

3.1 配線層構成と配線格子モデル

配線層数の多層化に応じて各配線層とレイアウト設計階層の対応関係を**表 1** のように設定した．三つの配線格子は，**図 2** (左)に示すとおり同一平面上に投影したとき，互いに重ならない構成をとる．層間接続は図 2 (右)のように，隣接する 2 層をスルーホールによって行う．

3.2 セルモデル

ブロック内自動配置の対象となるのが「セル」である．自動配置されるセルは形，端子座標が前もって定められた規則に即して設計されていることが必要である．**図 3** (A)にセルモデルを示す．



(A) Cell　(B) Block　(C) Block assembly　(D) Chip

図 3 レイアウトモデル
Fig. 3 Layout model.

【形状】 矩形とし，幅は，Poly Si 配線格子の整数倍で可変，高さは AL1 配線格子の整数倍で可変である．

【端子】 端子のX座標はセル原点に対して $2N+1$ ($N=0,1,2,\cdots$) の関係を保ちセルの上/下辺に置く．また，特殊な例として AL1 第0チャネル上にも端子をもちセルとセルを内部で配線することもできる．

3.3 ブロックモデル

ブロックモデルを図3(B)に示す．ブロック内部ではセルを中心線をそろえて横一列に並べ，こうしてできるセル列を縦方向に複数段配置する構成をとる．

【形状】 矩形であり，その高さ，幅はセル列段数に応じて可変である．ブロックの大きさ（高さ，幅）は自動配置配線の結果によって確定する．

【端子】 端子存在位置に内部散在型と周辺型の2種類があり，内部散在型の場合プログラムで端子位置を決定する．また，周辺型の場合，各端子に対し上/下/左/右，の辺指定を自動配線実行前に行う．

3.4 ブロックアセンブリモデル

2章で述べたとおり，ブロック間配線においてチップを複数個のブロック集合—ブロックアセンブリ—に細分割できる．分割に当たっては，チップ内部で各ブロックの絶対座標が確定していなければならない．設計が完了した各ブロックアセンブリをジグソーパズルのようにはめ合わせることによってチップが復元される．モデルを図3(C)に示す．

【形状】 矩形，もしくは矩形の各頂点が直角に欠けた差分矩形とする．

【端子】 ブロックアセンブリの各端子はいずれも分割境界上に置く．これらの端子はチップの細分割で一時的に発生したもので，レイアウト後の各ブロックアセンブリを結合すると消滅する．

3.5 チップモデル

チップは LSI 設計階層上，最上位に位置づけられるものであり，内部論理としてのブロックの集合および周辺部として入出力バッファとボンディングパッドが含まれる．図3(D)にチップモデルを示す．周辺部は自動レイアウトの対象としない．

4. 階層型レイアウト設計方式

本 VLSI の階層レイアウト設計における設計手順はトップダウンとボトムアップの2種類の手続きに大別される．トップダウン手続きでは設計の枠組を設定し，ボトムアップ手続きでは与えられた枠組の中で詳細化，最適化を図る．**図4**によってその手順を説明する．前提として，論理設計は終了し，ブロック分割および各論理要素のセル割当ては済んでいるものとする．

1) ブロックの相対配置

ブロックの相対配置を考えるとき，各ブロックの概略レイアウトイメージをつかんでおくことが効果的である．このために，ブロックを構成するセル列数をパラメータとしてブロック内自動配置配線を実行し，一つのブロックのアスペクト比が変化した‘レプリカ’を作成する．ブロックの相対配置はこの‘レプリカ’のなかから適当な一つを選択して行う．

各ブロック相互のデータ/制御の流れや LSI ピンを考慮してブロック間の相対位置関係を決める．これによりそれぞれのブロックにおいて外部端子の引出し方向，特定セルの配置位置などが明らかになる．これらの事項は次のブロック内部配置配線設計における制約条件となる．

2) ブロック内部の配置配線設計

各ブロックのレイアウトを行う．この結果ブロックのサイズが確定する．

3) ブロックの絶対配置

レイアウトが完了しサイズの確定したブロックをチップ上に配置し，各ブロックの絶対座標を決定する．このとき，チップサイズの設計目標値が拘束条件になる．

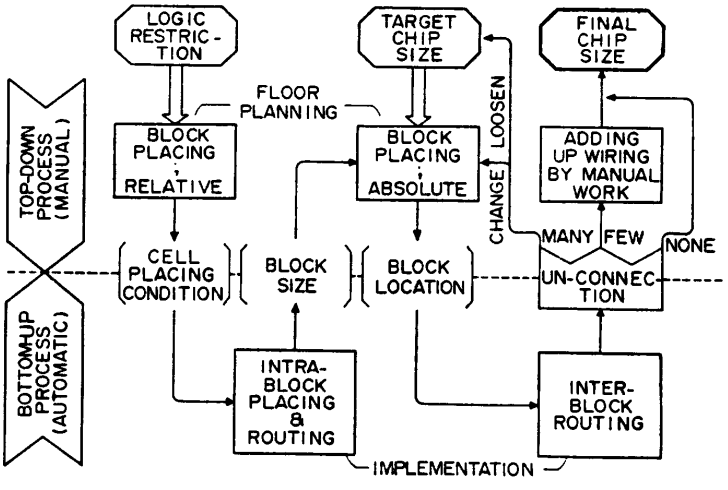


図4 階層型レイアウト設計フロー
Fig. 4 Hierarchical layout design flow.

4) ブロック間配線設計

チップ上での座標が確定しているブロックの外部端子間での配線を行う．複数設計者による分割並行設計を行う場合，チップをブロックアセンブリへ分割する．具体的には，チップの論理分割とアセンブリ境界における仮想端子位置の決定がなされることになる．

切り出されたブロックアセンブリでは境界の仮想端子位置と各端子のネット名が確定している．この結果各ブロックアセンブリは独立に自動配線可能となる．配線後のアセンブリを元の分割境界で結合することによりチップが復元できる．

5) 配線評価

ブロック間配線において配線領域の大きさが固定されているため，場合によっては領域内で配線が完結せず「未配線（あふれ配線）」が生ずることがある．その程度によって次の三つの対策をとる．

〈イ〉：未配線数が少なく人手による追加配線が可能な場合―人手による追加配線を行い設計を完結する．

〈ロ〉：未配線数が多い，もしくは，未配線は少ないが人手追加配線が困難な場合―チップ面積の条件は変えず，ブロックの配置を変更して再度4) を行う．

〈ハ〉：〈ロ〉においても配線が完結しない場合―目標としたチップ面積の拘束条件をゆるめ，再度4) あるいは5) を行う．

5. レイアウトプログラムの機能

4章で述べた設計方式を実現するための主要プログラムの機能について簡単に述べる．

1) フロアプラン

5章において1), 3) がフロアプランと呼ばれる機能の範囲に属する．フロアプランの基本機能は各ブロックのマクロな相対位置関係が与えられたときブロック間の配線経路を予測し，これに基づいた配線領域幅を推定してブロックの絶対配置座標を決定することである．このために具備すべき主要機能項目としてブロック間ネットの配線混雑解析があげられる．混雑解析指標としてここでは下式で「トータルトラック要求率：α」と「ローカルトラック要求率：β」を定義する．

α＝予測全配線長/使用可能全トラック長

β＝予測トラック数/使用可能トラック数

$\alpha \leqq 50\%$, $\beta \leqq 80\%$ を満たすようにブロックの位置およびチップサイズを調整する[8]．

2) ブロック内自動配置配線

セル間の論理結合関係を入力し3.3節で述べたレイアウトを生成するものである．ブロックの大きさは配置配線結果によって確定する．したがって可能な限り面積を小さくするように各セルの位置を求めることが重要である．ブロックを形成するセル列段数は自動配置配線の実行パラメータとして指定されるものとする．

3) ブロック間自動配線

フロアプランによっておのおののブロックのチップ内部での絶対座標が確定した後，ブロックアセンブリとして切り出された領域内でブロック間の自動配線を行う．

ブロック内自動配線とブロック間自動配線の違いは前者は配線によってブロックの大きさが確定するのに対して，後者は配線すべき領域の大きさが絶対条件として与えられていることである．このため場合によってはすべてのネットがこの領域で配線できず未配線を生じることがある．したがって，ブロック間自動配線の課題は未配線数の削減，言い換えれば自動配線率の向上である．

6. 自動レイアウトアルゴリズム

6.1 配置問題

ブロック内自動配置の基本問題は，ブロックの面積を最小とするセルの2次元配置位置を決定することである．図3(B)において，配線領域のX成分はセル列の長さで決まるため，面積を支配するのは領域のY成分，すなわち AL1 の配線トラック本数である．AL1のトラック本数を少なくするためには，結局，AL1配線（X方向配線）の長さを短くすればよい．事実，われわれの実測によればブロック面積と AL1 の全配線長には正の相関が認められる[9]．以上の考察からブロック内自動配置問題を次のように設定する．

「ブロック内セルの2次元配置を自動決定するに際して，目的関数Fを当該ブロックの全結線に関するX方向成分の総和とし，Fを最小とするようなセルの位置を求めること．」

自動配置処理は初期配置と配置改善の二つの処理よりなる．

6.1.1 初期配置

初期配置処理はセルの2次元配列を生成する手続きである．われわれはこの2次元配列生成に関していくつかの手法を検討し評価した（詳細は7.1節を参照されたい）．ここではそのなかで最良の結果を与えたクラスタリング2次元直接配置法について述べる．

【クラススタリング2次元配置法】

クラスタリング2次元配置は次の四つの手順で実行する.

(1) 配置セルのクラスタリング

(2) クラスタ Tree の形成

(3) Tree の頂点から下方に向かって Tree を分解しつつ各 Tree のノードをチップの部分領域にマッピングする. 部分領域への分割の順序は外部パラメータによって選択可能である.

(4) 部分領域間でクラスタを交換する. この場合クラスタのみならずセル単位の交換も可能である. なおセルの交換の場合には残りの部分についてクラスタ Tree を再度形成し(3)を繰り返す.

すべてのセルをマッピングすることによってセルの2次元配列が得られる.

6.1.2 配置改善

得られた2次元配置に対して, 各セル列単位に目的関数 F を最小とするべく個々のセルの位置を変更する. これは次の三つの手続きに要約される.

(1) 選ばれた一つのセルに関して最適移動位置を求める.

(2) 求めた位置に移動させることで目的関数 F の値が移動前よりも減少する場合に限り実際にセルを移動させる.

(3) すべてのセルが移動しなくなるまで(1)(2)を繰り返す.

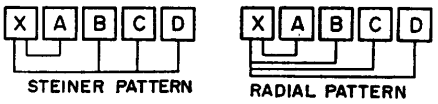
以上の手続きのなかで, (1)の移動最適位置を求める方法が問題となる. 本論文ではチャネル割当て配線によって作られる配線パターンと整合のとれた下記の手法を提案する.

【ネットバランス法】

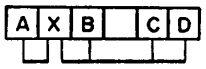
移動対象となったセルの最適位置を求めるには, 対象セル X に関してある距離関数を定義し, 一定条件下で最適解を得る処理が必要となる. チャネル割当て配線が生成する配線パターンは幹線支線方式に基づいた図5(A)で左に示す Steiner Tree となる. したがって, この場合, 6.1 節で述べた目的関数 F は幹線長の総和に対応する. このとき, 問題は次のように定式化できる.

命題: 数直線上 N 個の点(端子) $a_1, a_2, \cdots, a_n$; $a_i \geqq 0$ があり, それらは M 個のグループ(ネット) $S_1, S_2, \cdots, S_m$ に分割されている. また数直線上の点の有限集合 S に対して L(=ネット長) を次式で定義する.

$$L(S)=\max(S)-\min(S)$$



(A) 想定配線パターン



(B) 最適移動位置

図5 ネットバランス法
Fig. 5 Net balance placement.

ここに $\max(S)$, $\min(S)$ はそれぞれ S の元の最大値, 最小値であり. このとき, 一点 X(=セル) を座標 x に置いたとき,

$$f(x)=\sum_{i=1}^{m} L(S_i \cup \{x\}) \quad \text{が最小}$$

となる x を求めよ.

この命題の解となる移動最適位置は図5(B)に示すように,「セル X に関して右方向に拡がるネットと左方向に拡がるネットの数が等しくなる点」である(証明は付録を参照されたい). なおこのような点は, ネットの最左および最右点からなる集合,

$$P=\{\max(S_i), \min(S_i) \mid i=1, \cdots, m\}$$

において P の元を昇順(または降順)に並べたときの中央点に対応することは明らかである. この意味で本手法を「ネットバランス法」と名づけた.

6.2 配線問題

階層設計において通常とられる方法に, 上位階層内での設計時に下位階層をブラックボックス化し, その内部状態には関知しないという考えがある. しかし配線問題においてはこの考えは好ましくない. その理由はチップサイズを小さくする必要から, 設計階層にまたがってできるだけ配線層を共有し配線トラックの利用効率を上げることが要求されるからである. すなわち共有している配線層のあるトラックに着目したとき, 下位階層で部分的に使用された残りの部分は上位階層内の配線で使用できるような自動配線機能が必要となる.

本配線方式ではこのような要請および4章で述べた階層設計方法に整合するものとして, 二つの配線アルゴリズムを各設計階層に対応させこの問題を解決した.

6.2.1 ブロック内配線アルゴリズム

1) 両面端子の接続決定

セルは上/下辺に同電位端子をもつため, どちらの

端子を使用するかを決める．これは幹線のセル列間チャネル割付けに相当する．これには着目セル列の上下チャネルにおいて幹線の重なり分布を計算し，そのピーク値が小さい側のチャネルを選択する．

2)　Y方向配線層選択

Y方向配線に使用できる配線層は Poly Si 層と第2 AL 層の二つがある．第2 AL 層は主としてブロック間配線に用いるが，ブロック内配線においてもディレイが問題になるネットはその配線長に応じて第2 AL 層を選択的に用いる．このために，各ネットで出力端子から入力端子までのY方向距離 D_j を下式で推定し，設定されている閾値との大小比較結果をもとに使用層の割当て制御を行う．

$$D_j=\sum(H_k+C_m)$$

H_k：出力端子からそのネット内の入力端子 j に至るまでに縦断する配線領域 k の高さ

C_m：出力端子からそのネット内の入力端子 j に至るまでに縦断するセル列の高さ

H_k は1) の処理によって得られる各セル列間配線領域のX方向トラック数から求める．

3)　セル列横断点の決定

各ネットについてセル列上での垂直方向の横断点X座標を決定する．横断点は各ネットで1点/1セル列としネット内のピン座標分布でY座標の最小，最大区間内の1点を選択する．

4)　詳細配線

3)までの処理によって，セル列で挟まれた各配線領域内での上下対向端子の位置関係が確定する．ここでチャネル割当てアルゴリズムによって配線を行う．幹線は AL 1，支線は Poly Si と AL 2 である．

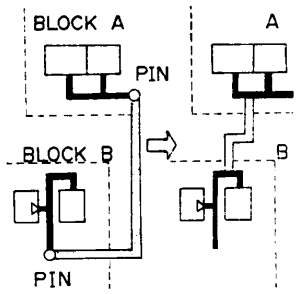


図 6　等電位配線
Fig. 6　Equi-potential routing.

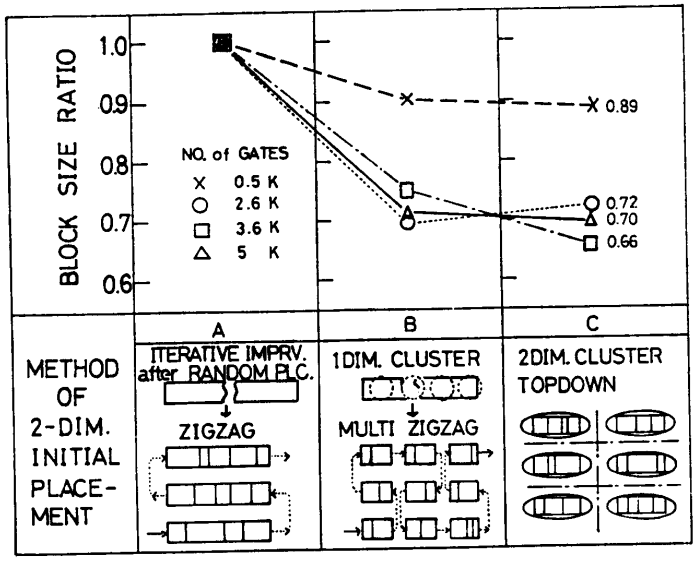


図 7　初期配置法の比較
Fig. 7　Comparison of initial placement procedures.

6.2.2　ブロック間配線アルゴリズム

ブロック間配線では配線トラックの利用率を上げるため，ブロック上空の配線を認める．それには，下位階層であるブロック内配線経路およびセル配置領域を詳細に調べ，これらを回避しながら配線経路を決定してゆく処理が必要となる．これを可能とするためにブロック間配線アルゴリズムとして Lee アルゴリズムを採用した．またその効果を十分引き出すためにブロックの端子はブロック内部に散在させる方式を採用した．

本配線プログラムはたんにブロック端子間を接続するのでなく，図6のように，その端子を含むブロック内既存配線パターン相互の接続機能（これを等電位配線と呼ぶ）を有している．8章で述べるチップの例では等電位配線により未配線が16本減少しその効果が実証された．

7. 性能評価

7.1　初期配置手法の比較

図7に初期配置手法の違いによるブロックサイズの比較を示す．(A)はセルの1次元順列を作成した後ネットバランス法による配置改善をし，最終の順列を保存したまま図のように折り曲げ（単純つづら折り）て2次元配置を得たものである．(B)は最終の1次元順列をクラスタ Tree の左右分解の繰返しによって作成し，これを図のように折り曲げ（多重つづら折り）て2次元配置とした．(C)は本プログラムで実現したクラスリング2次元直接配置法である．図より，(C)が全体として面積的に最も良好な結果を与えていること

表 2 配置改善法の比較
Table 2 Comparison of placement improvement procedures.

	A	B
BLK 1	1.86 mm²	1.87 mm²
BLK 2	1.13	1.16
BLK 3	1.19	1.22
BLK 4	1.08	1.09
BLK 5	1.56	1.56
Total	6.82	6.90

A: Net-balance method
B: Center of gravity method

がわかる.

7.2 配置改善法の比較

6.1.2 項で述べたネットバランス法と，従来用いられてきた重心法によるブロックサイズを評価した結果を表 2 に示す．初期配置はともにクラスタリング 2 次元直接配置法を用いた．表よりネットバランス法の優位性がわかる．

重心法では今回のブロックモデルの場合 $\{a_i\}$ をセル X につながるすべてのネットの端子座標の集合としたとき，$G(X)=\Sigma|a_i-x|$ を最小とする x を求める問題を解くことになる．この $G(X)$ は前掲図 5 (A) の右に示すようにセル X からそれに接続するすべてのセルに対して放射状の配線を想定した距離関数を意味しており，現実の配線パターンとの整合性に欠けることは明らかである．

7.3 処理時間

図 8 にブロック内自動配置配線の処理時間を示す．使用した計算機の性能は 3 MIPS である．5 K ゲート規模のブロックで配置に 102 秒，配線に 150 秒を要した．配置，配線時間はそれぞれゲート数の 1.4 乗，1.2 乗に比例している．

7.4 配線長の評価

2 章で述べたとおり大規模回路に対するレイアウト設計法として，われわれはブロックアセンブリ分割の考えを導入した．ここでは分割/一括の二つの設計法についてその設計品質を配線長の観点から実験的に考察する．図 9 は 6,000 ゲート規模の論理データに対し，チップを 4 ブロックアセンブリに分割してレイアウトした場合と，分割せずにレイアウトした場合の配線長の分布を示すものである．正規化配線長（チップの 1 辺長で正規化した配線長）が 1.0 以内で配線されているネットは分割設計では 97% であるのに対し一括設計では 96% となっている．また，最大ネット長

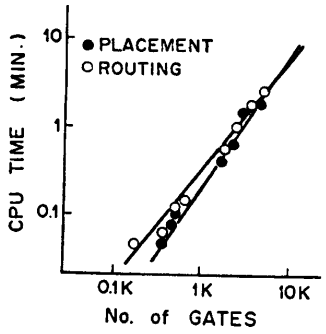


図 8 ブロック内配置配線処理時間
Fig. 8 Processing time of intra-block layout.

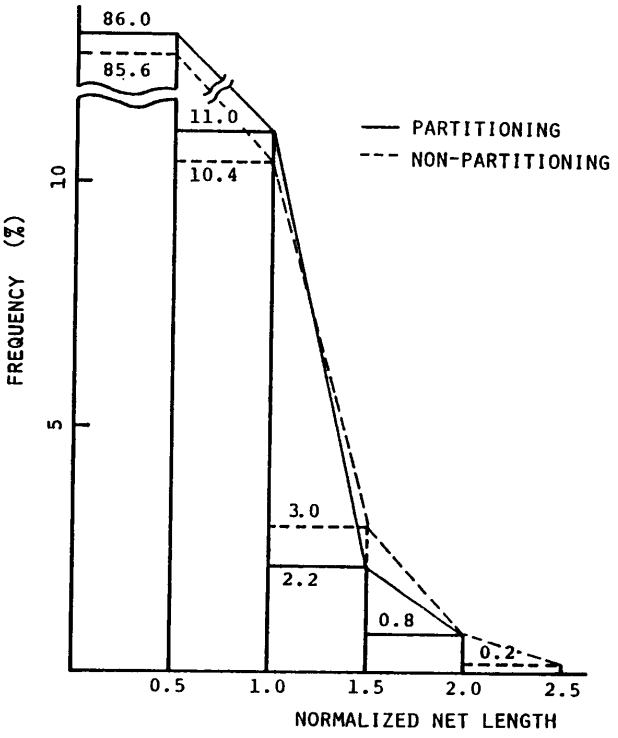


図 9 配線長分布
Fig. 9 Net length distribution.

に着目したとき，分割設計ではすべてのネットが正規化配線長 2.0 以下に納まっているが，一括設計では 2.0～2.5 に 0.2% 残っており，分布のすそが広がっている．この実験結果は分割設計が配線長の点で有利なことを示しているが，これは分割による配線領域限定の結果，一括設計の場合よりも迂回配線の発生が抑制されるためと推察される．

8. 設計事例

図 10 に本レイアウトシステムで設計された VLSI を示す．この VLSI は中型計算機クラスの性能をも

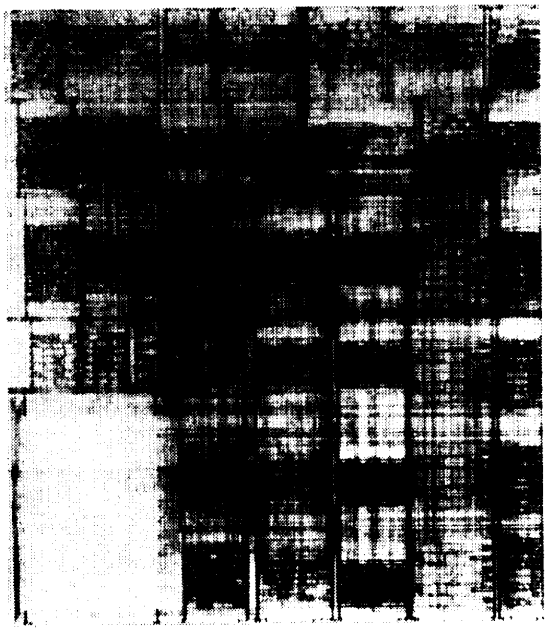


図 10 チップレイアウト結果
Fig. 10 An example of chip layout.

つ CPU 部であり 32 ビットの四則演算および種々の論理演算機能をもつ．チップの諸元を下記に示す．

プロセス	2 μCMOS
チップサイズ	12 mm×12 mm
トランジスタ数	74,000
ゲート数	17,000
セル数	5,700
RAM	2 k ビット
パッケージピン数	200

このチップは四つのブロックアセンブリに分割して設計した．ブロック間配線本数は全アセンブリ合計で 2,726 本，このときの処理時間は 256 分，未配線数は 9 本であった．またトラック利用率は AL 1，AL 2 それぞれ 47.0%，49.1% であった．本チップは 10 人月以下の工数，設計エラー 0 でレイアウトされ第一回試作においてすべての正常動作を確認した．

9. むすび

階層型設計を指向したカスタム VLSI のレイアウト方式および自動配置配線プログラムの機能，性能について述べた．設計対象規模の増大と期間短縮の要請に効果的に対応するため，ブロック集合を任意に分割し並行設計を可能とするブロックアセンブリの概念を導入した．自動配置ではチャネル割当て配線と整合性のよいネットバランス法を新たに提案，実施した．また自動配線では設計階層に応じてチャネル割当て法と Lee アルゴリズムの二つを使い分け配線層数の多層化に適した自動配線方式を実現した．

本プログラムを用い 20 k ゲート級の VLSI を設計不良 0 でレイアウトすることができた．今後の課題としてはフロアプランにおける設計効率の向上，たとえばブロックの自動配置機能の実現が挙げられる．

謝辞 日頃，研究推進に当たって種々のご配慮をいただく，堀越彌部長，大野拳廣部長，谷口研二部長に深謝します．また，プログラム開発にご尽力いただいた，早瀬道芳，石井建基，三浦地平，小川泰，杉山俊樹，岸田邦明，佐藤康夫の諸氏に深謝します．

参考文献

1) Shiraishi, H. et al.: Efficient Placement and Routing Techniques for Master Slice LSI, Proc. of 17th DAC, pp. 458-464 (1980).
2) 寺井他：マスタスライス LSI の配置配線プログラム，情報処理学会設計自動化研資 EDD 81-9, SC 81-9, pp. 31-40 (1981).
3) Yabe, S. et al.: A Hierarchical Layout System for Gate Arrays, Proc. of ICCAD, pp. 46-48 (1983).
4) 寺井他：大規模 CMOS マスタスライス自動配置配線方式，信学技報 CAS 84 (1984.2).
5) Horiguchi, S. et al.: An Automatically Designed 32 bit CMOS VLSI Processor, Proc. of ISSCC, pp. 54-55 (1982).
6) 吉村他：論理 VLSI の階層的自動レイアウト法，信学技報 SSD81-54, pp. 23-30 (1981).
7) Sato, K. et al.: An Integrated Custom VLSI Design System, Proc. of ICCC, pp. 516-519 (1982).
8) Ohno, Y. et al.: Integrated Design Automation System for Custom & Gate Array VLSI Design, Proc. of ICCC. pp. 512-515 (1982).
9) Kozawa, T. et al.: Automatic Placement Algorithms for High Packing Density VLSI, Proc. of 20th DAC. pp. 175-181 (1983).

付録　ネットバランス法の証明

セル X を数直線上の点 x においたときグループ S_i に対する関数 $L(S_i \cup \{x\})$ は x と $\max(S_i)$, $\min(S_i)$ の関係によって次の(1)～(3)の3ケースのいずれかとなる．

$$L(S_i)+(x-\max(S_i)), \quad \text{if } x>\max(S_i) \tag{1}$$

$$L(S_i), \quad \text{if } \min(S_i)\leqq x\leqq \max(S_i) \tag{2}$$

$$L(S_i)+(\min(S_i)-x),\quad \text{if } x<\min(S_i) \tag{3}$$

いま，

$$L_x=\{i\mid\max(S_i)<x\} \tag{4}$$

$$R_x=\{i\mid\min(S_i)>x\} \tag{5}$$

なる集合 L_x, R_x を定義する．L_x はxより左にあるネット，R_x はxより右にあるネットに対応する．

(1)～(5)を考慮して $f(x)$ を展開すると，

$$\begin{aligned}f(x)&=\sum_{i=1}^{m}L(S_i\cup\{x\})\\&=\sum_{i=1}^{m}L(S_i)+\sum_{i\in L_x}(x-\max(S_i))+\sum_{i\in R_x}(\min(S_i)-x)\\&=\sum_{i=1}^{m}L(S_i)+(|L_x|-|R_x|)x-\sum_{i\in L_x}\max(S_i)+\sum_{i\in R_x}\min(S_i)\end{aligned} \tag{6}$$

いま，$|L_x|=|R_x|$ となるxの値を x_0, $|L_x|\neq|R_x|$ となる一つのxを x' ($x'>x_0$) とする．このとき x' について(4)(5)式は

$$L_{x'}=L_{x_0}\cup\{i_1\cdots i_p\} \tag{7}$$

$$R_{x'}=R_{x_0}-\{j_1\cdots j_q\} \tag{8}$$

と書ける．(7)(8)式は $x=x_0$ において $|L_x|=|R_x|$ が成立していたのが $x=x'$ へと移動することによって“その最大値が x' より小さなネット”が L_{x_0} からp個増え，逆に，R_{x_0} のなかで“その最小値が x' より大きくないネット”がq個出現し，これらを R_{x_0} から除いたものが $R_{x'}$ になることを表す．このとき，

$$\begin{aligned}&f(x')-f(x_0)\\&=\sum_{i=1}^{m}L(S_i)+(|L_{x'}|-|R_{x'}|)x'-\sum_{i\in L_{x'}}\max(S_i)+\sum_{i\in R_{x'}}\min(S_i)\\&\quad-\sum_{i=1}^{m}L(S_i)-(|L_{x_0}|-|R_{x_0}|)x_0+\sum_{i\in L_{x_0}}\max(S_i)-\sum_{i\in R_{x_0}}\min(S_i)\\&=(|L_{x_0}|+p-|R_{x_0}|+q)x'-\sum_{k=1}^{q}\min(S_{j_k})-\sum_{r=1}^{p}\max(S_{i_r})\\&=\sum_{k=1}^{q}(x'-\min(S_{j_k}))+\sum_{r=1}^{p}(x'-\max(S_{i_r}))\end{aligned} \tag{9}$$

ここで(7)(8)の解釈から

$$x'\geqq\min(S_{j_k})$$

$$x'\geqq\max(S_{i_r})$$

したがって，$f(x')-f(x_0)\geqq 0$.

すなわち $|L_x|=|R_x|$ となるxが最適解である．

(証明終り)

(昭和59年2月14日受付)
(昭和59年7月19日採録)