

# LSIレイアウト設計システムSHARPSについて

千葉 徹 奥田 登 神戸 尚志 西岡 郁夫  
 犬伏 恒雄 木村 征二 (シャープ株式会社)

## 1. はじめに

近年、電子ビーム露光装置の開発を始めとする製造技術の急速な進歩によって、LSIの高集積化が進むと共に、ユーザの機能に対する要請が多様化しLSI開発における設計プロセスが複雑化・長期化してきている。これに対応するためには、LSIの設計自動化技術を導入し、LSIの開発期間の短縮と共に、設計品質の確保をはかることが、不可欠である。

LSIの自動設計に関しては、従来から多くの研究が行われており、マスタライズLSIではシミュレーションを含めた自動設計システム<sup>[1]</sup>が既に実用段階にある。また、多様な外形・大きさを持つ機能ブロックの組み合わせで構成されるLSIに対しても自動配置・配線システム<sup>[2,3]</sup>の開発が行われて来た。

我々は、現在、任意の形状・大きさを持つ機能ブロックからなるLSIについて、階層的設計を実行できるレイア

ウト設計システムSHARPS (SHARP Hierarchical Automatic Routing and Placement System)<sup>[4]</sup>を開発中である。本文では、SHARPSの概要をソフトウェア構成を中心に報告する。

## 2. システム構成

一般に、LSIの設計プロセスのなかでレイアウト設計に注目した時、CADシステムが持つべき特徴として次のものがあげられよう。

- (I) 設計フローの広範囲を包含する自動設計機能をもつ。
- (II) 個々の自動設計機能の能力改善が絶えず行えるよう、各プログラムモジュールがレイアウトデータベースを取り巻くように配置され、相互のコミュニケーションはデータベースを介して行なう。
- (III) 人間の創造的な能力を引き出すために、設計フローのどの時点においても会話型の介入ができる。

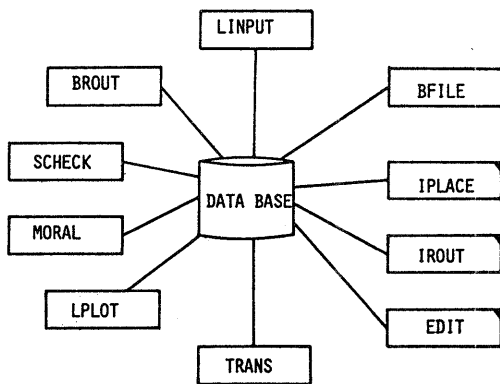


図1. ソフトウェア構成

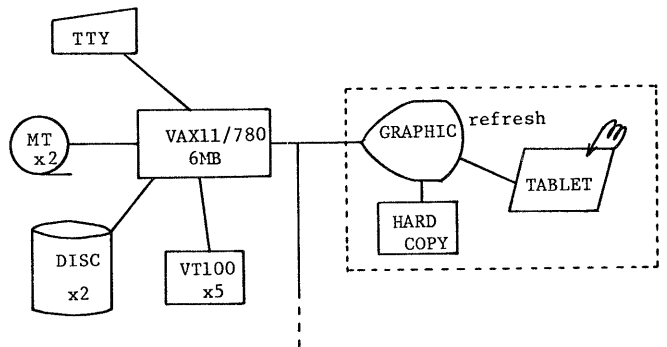


図2. ハードウェア構成

(Ⅳ) 設計の無謬性を追求するために、人間の介入は大局的な判断にとどめ、個々のデータの修正はシステムの自動機能が行なう。

(Ⅴ) 膨大なデータ量の問題を克服するため、階層的な設計システムとする。SHARPSはこのような観点から設計されており、そのソフトウェア構成とハードウェア構成を図1,2に示す。

SHARPSでは、いくつかの機能ブロックの設計から始めて、形状・大きさの異なる機能ブロックを組み合せながら、上位階層の機能ブロックを構成し、これを繰り返して、チップ全体の設計を行なう(図3)。SHARPSは、このような階層的設計法を効率よく実行するために、入出力処理プログラム、自動配置プログラム、自動配線プログラム、ランダム論理部自動レイアウトプログラム等のバッチ処理型プログラムと共にグラフィックディスプレイを用いた会話型半自動修正プログラムを備え、これらがデータベース(マスタファイル)を中心に配置された融通性に富んだシステムになっている。

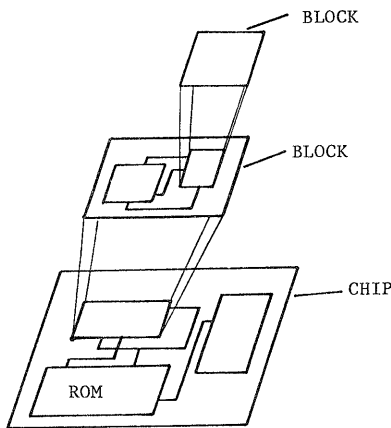


図3. 階層構造

SHARPSのデータベースは、ブロックファイルとパターン情報ファイルの二つのファイルから成る。図4-a, bにこれらのデータ構成を示す。

ブロックファイルには、ブロック固有の情報とブロック間の階層構造(木構造)を表現する情報が蓄えられており、パターン情報ファイルには、各ブロックに含まれる、配線・I/O端子・コンタクト・トランジスタ・外形等の情報が蓄えられている。

- ブロック識別名
- 層情報
- パッド情報(パッド・ブロックの場合)
- 論理格子の大きさ
- 他ブロックとの階層構造情報
- 配置情報(他のブロック内に配置されている場合: 位置・方向)
- コピー情報(反復使用される場合)
- 当該ブロックに含まれるレイアウトパターン情報を指すポインタ群  
など

図4-a. ブロックファイルのデータ構成

- 配線・I/O端子・コンタクト・外形トランジスタ等の区別
- 信号名
- 座標値
- 層情報  
など

図4-b. パターン情報ファイルのデータ構成

## 2-1 LINPUT (入カプログラム)

LINPUTは、機能ブロックをデータベースへ入力するためのプログラムである。図4に示されたデータは、簡単なLINPUT言語で記述されバッチ処理されるだけでなく、端末から会話的に入力・修正することが出来る。また、LINPUTは、Applicon 図形処理システムに登録されている機能ブロックを、SHARPSデータベースへ入力する場合のインターフェースとしても働く。

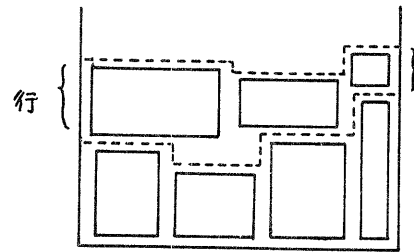


図5. 行毎の配置

## 2-2 IPLACE (配置プログラム)<sup>[5]</sup>

IPLACEは機能ブロックの(初期)配置を行なうプログラムである。本システムでは、あらゆる形状の機能ブロックを入力として許しているが、これを唯一つのアルゴリズムで対処しようとするれば、極めて非効率的であることが予想される。従って、幾つかの配置手法を用意しておき、入力の型によって使い分けるという方法を用いている。例えば、ほぼ同じようなセルの並んだいわゆるポリセル方式のLSIの場合、

(1) 面積の評価

(2) セルのクラスタリング(各行に対してさせてセルを束ねる)

(3) 各行毎、各行内のセルの一次元配置

(4) 配置改善

と言った手順で対処する。

ところで一般に、不規則なサイズ・形状を持つ矩形で表わされる機能ブロックの配置問題は、配線要求と各ブロックのサイズ・形状を同時に評価する必要があるため、極めて難しい問題である。これに対する解法の一つに Min-Cut 法<sup>[6]</sup>があるがこの方法は、分割された各部分間の結線要求を最小にしようとするだけでなく、同時に各部分に含まれるブロックの総面積を見積り、それらのバランスをとることが出来、異なるサイズのブロックがある場合に

有効であると思われる。しかし、形状がかなり異なる場合には、ブロックの絶対位置を決める段階に困難があろう。

IPLACEは、いわゆる構成的手法を基にしたプログラムであり、Min-Cut法を実行した後の絶対位置決定時に使うこともできる。図5に示されるように、親ブロックの一边から始めて、既に配置されているブロックや外部端子に対する配線要求の強さを考慮しつつ、ある定められた横巾のなかに一つの行として配置すべきブロックを選び出し、次に配線領域を見積りながら高さや凹凸が最小になるようにブロック配置を決定する。以上の操作をくりかえして、全体の配置を求める。

しかしながら、機能ブロックの形状・サイズに甚しい差がある場合には上記の手法も効率的でないことがあり、このために、IPLACEは会話的な機能を有している。ユーザは、自動配置する機能ブロックの個数Nを指定でき、N個ずつ自動配置された状況をグラフィックディスプレイで見、それを受理するか否かを決定できる。更に、その配置の変更も会話的にでき、途中でNの値を変更することも可能である。

IPLACEの概略フローを図6に示す。自動配置した後、後述の配線プログラムを適用した結果を図10に示す。

## 2-3 BROUT (配線プログラム)

ブロック間の配線を実行するバッチ処理型プログラムである。

自動配線の手法には、迷路法や線分探索法など<sup>(9)</sup>があり、プリント基板用にはよく使われているが、LSIでは最小面積で100%配線を実現するのが目標となるため、これらの手法をそのまま採用することはできない。

BROUTは、チャンネル配線手法<sup>(10)</sup>に基づくプログラムであり、LINPUTやIPLACEによって得られたブロック配置を入力として、ブロック間の配線要求を実現しながら、ブロックの絶対位置をも並行して決めて行く。入力ブロック配置は、相対的なものとみなされる。

BROUTの処理手順を以下に示す。

- ①チャンネルの交差状況の解析  
ブロック間の配線領域をチャンネルと呼ぶ(図7)。入力ブロック配置からチャンネルを求めた後、チャンネル間の交差状況およびチャンネルとブロック間の隣接関係を調べる。
- ②チャンネル順序制約グラフの作成  
T字交差している二つのチャンネルがある時、チャンネル4は7より先に設計されなければならない<sup>(11)</sup>(図7)。チャンネル順序制約グラフとは、チャンネルを頂点に対応させT字交差している時かつその時に限って、辺を付加したグラフである(図8)。

このグラフにループが存在しない時  
③へ行く。ループが存在する場合にはチャンネル配線順序を一意に決定できないため、入力ブロック配置を自動的に修正して、①へもどる。

### ③ネット情報の作成

配線要求は、ブロックのI/O端子に付与された信号名によって表わされており、I/O端子が同一信号名を持つ時同電位に配線される。ここでは、I/O端子の信号名よりネットを作成する。

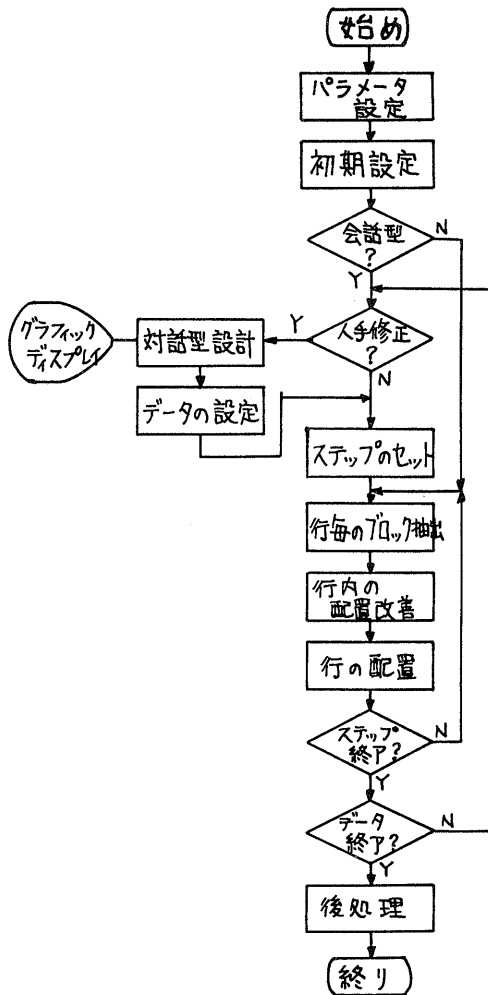


図6. IPLACE 概略フロー

#### ④束の認識

最上位階層ブロック(チップ)の場合配線要求には一定の規則性があると言われ,いくつかの配線を一つのまとまりとして,束状に配線するのが有効と思われる。ここでは,ネット情報を分析して,束として取り扱うべき配線を抽出している。

#### ⑤グローバル配線

ここでは,チップ面積を減少させる観点から,各配線要求に対して,経路すべきチャンネルの系列を求める。まず(サブ)チャンネルを頂点に対処させ,(サブ)チャンネル間の隣接関係を辺に対処させて配線グラフを構成する。次

に頂点および辺には,(サブ)チャンネルの長さ・配線混雑度に依存した重みを付与した配線グラフ上で迷路法探索を実行してグローバル配線を行う。

#### ⑥チャンネル配線

チャンネル内配線は, left-edge アルゴリズムに基づいている。配線すべきチャンネルCに隣接する二つのブロックB1, B2 について,チャンネル幅が最小になり,かつ未配線ネットが生じないように相対位置(チャンネルに並行な方向)を決めた後,配線(トラック割当)を実行する(図9)。トラック割当に際しては,ブロックB1, B2の形状の凹凸を有効利用すべく工夫がこらされている。

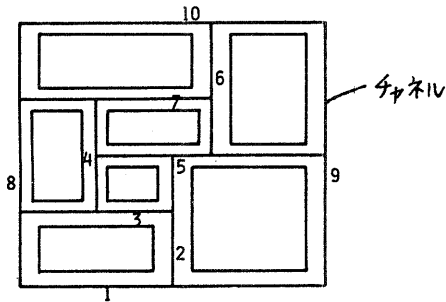


図7. チャンネル構造

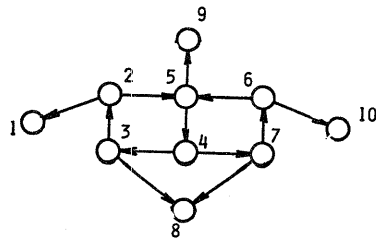


図8. チャンネル順序制約グラフ

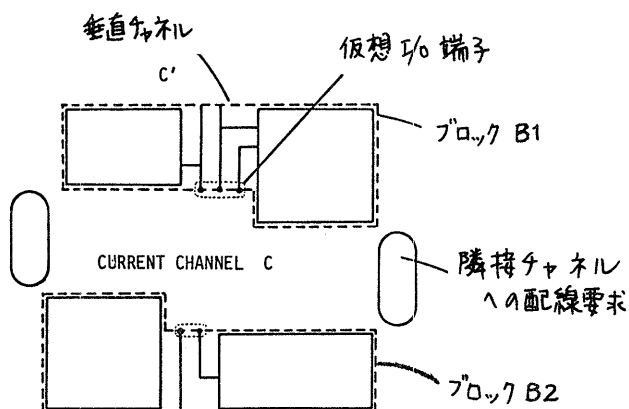


図9. チャンネル配線

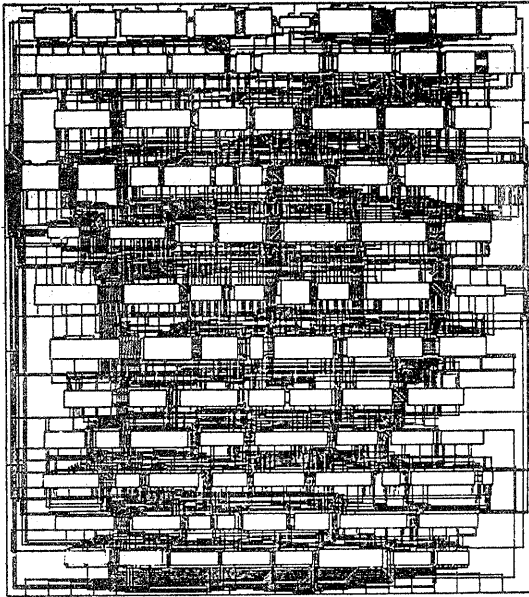


図10. BROUT配線例 (No.1)

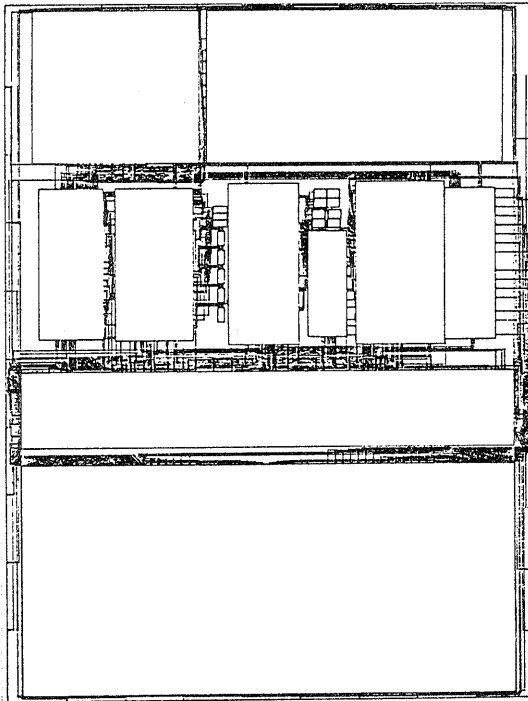


図11. BROUT配線例 (No.2)

### ①絶対座標の決定

一つのチャンネル内配線が実行されると、二つのブロックがマージされ新しい一つのブロックが生成される。この新しいブロックに含まれる配線・コンタクト・ブロックの相対座標は一意に定まるから、このようなマージ操作を繰返して、チップ全体が一つのブロックになった時、すべてのデータの相対位置が一意に定まることになる。ここでは、データ間の相対位置から、それらの絶対座標を計算する。

BROUTの実行結果は、再びBROUTへの入力として繰返し用いることができる。これによって、チャンネル長巾をより正確に見積ることになり、グローバル配線の精度が向上すると期待される。

二種のLSIに対するBROUTの配線例を図10, 11に示す。

### 2-4 IROUT (会話型自動再配線プログラム)<sup>[9]</sup>

BROUTの実行後、あるいは人手修正後に、その配線結果に対し、以下のような理由で配線を変更した11場合が生じる。

- 1) 配線の一部を変更して、面積の縮小をはかる。
- 2) デザインルールの変更に伴って、配線を変更する。
- 3) 一つの配線の変更に伴い、その近傍の配線についても経路変更した11。

このような修正が、配線混雑部分で必要となった場合、すべての作業を人手で行なうことは、大変な労力であり、かつ誤りを生み出しやすい。このために、変更した11場所、あるいは配線した11ネットを指定すただけで自動的に誤まりなく再配線する機能を持つプログラムは修正作業の効率化に役立つものと期待される。

IROUTの機能を以下に列挙する。

(1)指定された二点間の自動配線

タブレットによる位置指定，もしくはキーボード等からの座標入力された二点を自動配線する。配線アルゴリズムは，SoukupのFast-mageアルゴリズム<sup>[10]</sup>に基づくもので，経路探索時間の短縮化をはかった。これによると，経路が存在する場合，その一つが効率よく見い出される(図12)。

(2)指定されたネットの自動配線

ネットは一般に多端子なので，配線順序が総配線長に大きく影響する。ここでは，まずマンハッタン距離の最も大きい二点を選り出して配線し，次に未配線端子を順次取り出し，既配線の最も近い点を目標点として配線している。

(3)自動配線時の配線領域の限定

自動配線の対象となる配線領域のなかで，配線可能領域，あるいは配線禁

止領域を自由に設定できる。

(4)配線経路の変更

他の配線を可能とするためや，ブロックの配置を改善するために，配線の一定区間を指定した時，指定した方向へ自動的に凸状にまわり道させる。

(5)配線層の変更

他の配線を容易にしたり，配線経路の改善のために，既配線について層と点を指定した時，その点の近傍の配線層を自動的に変更する。この場合必要ならばコンタクトが自動的に発生される。

(6)平行線の付加

データ線やアドレス線などは束状にまとめて平行配線するのが望ましいことがある。基本線・平行線の本数・間隔を入力した時，自動的に平行線が付加される。

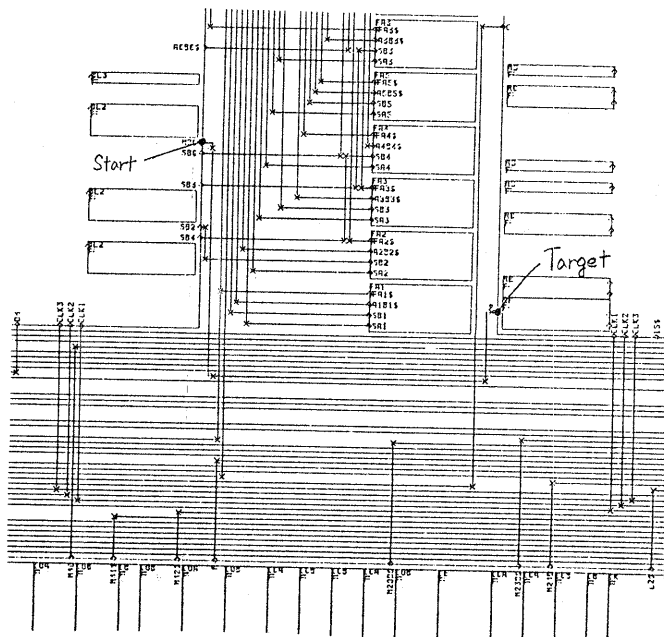


図12. IROUT配線例

## 2-5 MORAL (ランダム論理部レイアウトプログラム) <sup>[11]</sup>

MORALは、複合ゲートと呼ばれる多相クロックを用いたMOSレシオレス回路から成るランダム論理部に対して、レイアウトパターンを自動生成するプログラムである。

複合ゲートは負荷デバイスと入力デバイスから成る。負荷デバイスはプリチャージ用のトランジスタから成り、次段に接続されるインバータなどと共に一つのセルにまとめられ、人手により設計される。一方、入力デバイスは数個の相のクロックを入力とするトランジスタと論理関数を実現するための論理ブロックからなる(図13-a, b)。ランダム論理部は、図14に示すようにいくつかの複合ゲートを組み合わせて実現される。

MORALは、複合ゲートによって実現されるべき論理関数の集合を入力として、ランダム論理部の面積が最小となるように、複合ゲートの一次元配列の順序および信号線の水平トラックへの割当てを適切に定めて、各複合ゲートのレイアウトパターンを実現するものである。処理内容は次のようである。

### (1) 論理関数の単純化

個々の複合ゲートを最小面積で実現するために、入力された論理関数に対し

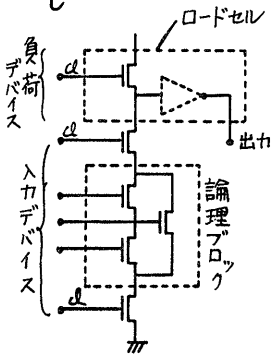


図13-a. 複合ゲート

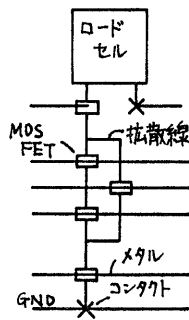


図13-b. レイアウトパターン

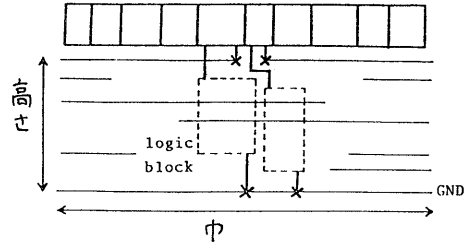


図14. ランダム論理部

### (1) 論理関数の単純化

個々の複合ゲートを最小面積で、実現するために、入力された論理関数に対し、積項の個数を最小にするように部分的にいくつかの手続を施す。

イ) べき等律 ( $X+X=X$ ,  $X \cdot X=X$ ) による単純化

ロ) 相補律 ( $X \cdot \bar{X}=\phi$ ) による単純化

ハ) 吸収律 ( $X+X \cdot Y=X$ ,  $(X+Y) \cdot X=X$ ) による単純化

ニ) 因数分解による単純化

### (2) 複合ゲートの配列順序の決定

ランダム論理部の高さは、複合ゲートの配列順序によって左右される。ここでは、クラスト成長法によって初期配列を求め、これに対してゲートの置換(単位操作)に基づく反復改良を施し近似解を求めている。

### (3) 信号線の水平トラックへの割当て

複合ゲートの配列順序が決定された後、各信号線を水平トラック上に割り当てなければならぬ。信号線の上下の位置関係は論理部の中に影響する。ここでは、上方のトラックから順次割り当てて初期解を求め、隣接ペア交換法で改良している。

### (4) レイアウトパターンの描画

ゲートと信号線の配列順序に基づいて、各論理関数を実現すべく、左方のゲートから順次既設計パターンの包絡線に沿わせるようにして、レイアウトパターンを描画する。

図15に出力結果の一例を示す。



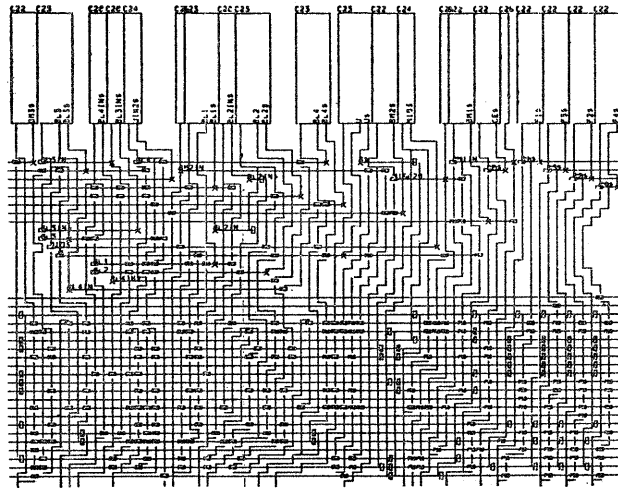


図15. MORAL 出力例

### 2-6 EDIT (人手修正)

グラフィックディスプレイ (リフレクシユ型) を用いてレイアウトパターンデータを人手で修正するためのプログラムである。EDIT は、グラフィックディスプレイにレイアウトパターンの一部を表示し、データの ADD, SELECT, MOVE, DELETE, MODIFY 等の機能を備えており、これらのコマンドとデータをキーボード、スタイラス、ジョイスティック等を用いて選択し、修正を施すことができる。

### 2-7 CHECK (パターンチェック・プログラム)

人手修正による誤まりをチェックするパターンチェックプログラムである。

- 1) 配線の開放・短絡チェック
- 2) 配線の容量・抵抗値のチェック
- 3) 各データの層チェック
- 4) ブロック間階層関係のチェック

### 2-8 BFILE (データベース・バックアップ・プログラム)

指定されたブロックのデータをディスク又は磁気テープへバックアップする。

- 1) データベースからバックアップファイルへのストア。
- 2) バックアップファイルからデータベースへのロード。
- 3) データベースの初期化
- 4) バックアップファイルの情報表示等の機能を持つ。

### 2-9 L PLOT (プロッタ・出力のための NC テープ作成)

データベース中のブロック名を指定して、そのレイアウトパターンを (ALCOMP, XYNETICS 等のペンプロッタ) へ出力するためのプログラムである。

### 2-10 TRANS (図形処理システムへの変換プログラム)

SHARPS データベースから図形処理システム (APPLICON) へのデータ変換を実行するプログラムである。

SHARPS は、システムの汎用性を高めるためにプロセスに依存しないシンボルレイアウトを行っている。そのため製造プロセスに基づいたコンポジットデータの生成は、図形処理システムによって行われるのである。

### 3. まとめ

本文では、LSIのレイアウト設計システム：SHARPSの概要を、ソフトウェア構成を中心に紹介した。SHARPSの特徴は、

- 1) 階層設計手法の採用により大規模集積回路の設計自動化を目指している。
- 2) 任意のサイズの機能ブロックの扱いを許しており、設計の自由度が高い。
- 3) 共通データベース(マスタファイル)の周囲に多くのバッチ型自動設計プログラムと共に、会話型プログラムを備え、人手の介入による柔軟な設計を実行できる。
- 4) レイアウトはシンボルを用いて行なっているため、扱いやすく、かつ図形処理システムへの変換プログラムによってどのような製造プロセスにも対応できる。

しかし、SHARPには未だ多くの改良すべき点が残されており、またレイアウトパターンのコンパクト化やシミュレーション等多くの機能の追加を必要としており、現在開発を進めている。

### 謝辞

本システムの開発にあたり、プログラミングに協力して頂いた集積回路事業部 西本盛史、林正樹の両君に深謝します。

### 文献

- [1] Shiraishi, H., Hirose, F., "Efficient placement and routing techniques for master slice LSI," Proc. 17th Design Automation Conference, 1980, pp. 485-464.
- [2] Kani, K., Kawanishi, H., and Kishimoto, A., "ROBIN: A building block LSI routing problem," Proc. International Symposium on Circuits and Systems, 1976, pp. 658-661.

- [3] Preas, B. T., "Placement and routing algorithm for hierarchical integrated circuit layout," Doctoral Dissertation, Stanford University, 1979.
- [4] Chiba, T., Okuda, N., Kambe, T., Nishioka, I., Inufushi, T., and Kimura, S., "SHARPS: A hierarchical layout system for LSI's," National Convention Record, IECE Japan, (to be appeared in March 1981).
- [5] Kambe, T., Chiba, T., Nishioka, I., Inufushi, T., and Kimura, S., "An initial placement of SHARPS," National Convention Record, IECE Japan, (to be appeared in March 1981).
- [6] Breuer, M. A., "Min-Cut placement," Journal of Design Automation and Fault tolerant Computing, vol. 1, no. 4, 1977, pp. 343-362.
- [7] Breuer, M. A., ed. Design Automation of Digital Systems, vol. 1: Theory and Technologies, Englewood Cliffs, N.J.: Prentice Hall, 1972.
- [8] Hashimoto, A. and Steven, J., "Wire routing by optimal channel assignment within large apertures," Proc. 8th Design Automation Workshop, 1971, pp. 158-169.
- [9] Kambe, T., Chiba, T., Okuda, N., Nishioka, I., Inufushi, T., and kimura, s., "A routing program supporting the interactive facilities for hierarchical LSI design system," Monograph, IECE, CAS 80-51, 1980, pp. 55-61.
- [10] Soukup, J., "Fast maze router," Proc. 15th Design Automation Conference, 1978, pp. 100-102.
- [11] Shirakawa, I., Okuda, N., Harada, T., Tani, S., and Ozaki, H., "A layout system of random logic portion in MOS LSI," Proc. 17th Design Automation Conference, 1980, pp. 92-99.