

ゲートアレイ用シンボリックレイアウトシステム: HSYLA

田代 雅久 川村 弘哉 小池 豊

沖電気工業(株) 超LSI開発センタ

ゲートアレイのライブラリ設計を支援する階層シンボリックレイアウトシステム HSYLA について報告する。HSYLA は、デザインルールに固定なゲートアレイのリーフセルをセルシンボルとして扱い、設計者は各セルシンボル中に含まれるトランジスタの端子間をシンボリック配線する事でライブラリの設計を行う。また、既に作成済みのシンボリック配線データを新規デザインルールのリーフセルに従って再配置する機能を持ち、デザインルールの更新時に過去の設計資産を再利用できる。これまでに250種類のライブラリ設計を行い、設計期間を従来の1/2に短縮した。

A Hierarchical Symbolic Layout System
for Gate-array Library Design

Masahisa TASHIRO Hiroya KAWAMURA Yutaka KOIKE

VLSI Research & Development Center
OKI Electric Industry Co., Ltd.
550-1, Higashiasakawa-cho, Hachioji-shi, Tokyo 193, Japan

This paper presents a hierarchical symbolic design system named HSYLA developed to support gate-array library design. HSYLA handles design rule dependent leaf cells as cell symbols. Designers do symbolic routing between cell symbol terminals using contact symbols, terminal symbols, and wire symbols. When the design rule was updated, HSYLA can automatically regenerate a symbolic diagram according to terminal positions of the new leaf cell symbols.

We designed about 250 kinds of gate-array libraries using HSYLA, and the design time was reduced by a half.

1.はじめに

ゲートアレイの大規模化が進み、標準ロジックで組んでいたシステムの置き換え需要のみでなく、従来スタンダードセルやフルカスタムでなければ対応できなかった製品を、ゲートアレイで実現する例が増加している。また、高速、高駆動能力のLSIを必要とする分野では、BiCMOSやECLゲートアレイへの要望も高まってきた。このような需要動向に対処するため、メーカー側では各種テクノロジーのゲートアレイのシリーズを用意し、各シリーズ毎に多種多様なライブラリを提供する必要に迫られており、ゲートアレイライブラリ設計工程の効率化が急務となっている。一方、LSI製造技術の急速な進歩に追従するためには、デザインルール更新時に設計済みのライブラリ資産を再利用できるようなライブラリ設計手法が不可欠である。

本稿では、上記の観点からゲートアレイのライブラリ設計専用開発した、階層シンボリックレイアウトシステムHSYLAについて述べる。HSYLAは、各種テクノロジーを扱うことが可能であり、また、デザインルールに独立なシンボリックレイアウト手法を導入することにより、ライブラリ設計資産の継続性を実現した。HSYLAを、SSIからMSIクラスまでのライブラリやRAM、ROMなど、250種のゲートアレイライブラリ設計に適用し、設計期間を従来の1/2に短縮することができた。

以降、HSYLAの特徴、システム構成、HSYLAによるライブラリ設計フロー、適用例を述べる。

2.HSYLAの特徴

本章では、各種テクノロジーを扱い、デザインルールに独立な階層シンボリックレイアウトシステムHSYLAの特徴につい

て述べる。

2-1.シンボリックレイアウト

ゲートアレイのレイアウトライブラリ設計は下地のリーフセルが固定でかつリーフセル間の配線のみであることから、依然としてレイアウトエディタを使用した人手設計手法を採っている場合が多い。これは、デザインルールが変わる度に多種多様なレイアウトライブラリを再設計することを意味し、設計生産性が悪い。

そこで、HSYLAはレイアウト言語とシンボリックレイアウトを併用した手法を採用し、デザインルールに独立なレイアウトライブラリ設計を可能にした。レイアウト言語は、シンボル化したリーフセルの配置に、シンボリックレイアウトは、リーフセル間の配線に使用する。この手法により、設計者は基本的なレイアウト設計の知識を持っていれば、特にデザインルールを意識しなくてもレイアウトライブラリの設計が可能となった。

2-2.各種テクノロジーの扱い

ゲートアレイは、素子のみを基本要素としたリーフセルがチップ全面にあらかじめ配置・製造されており、レイアウトライブラリにはリーフセル中の素子間を接続する配線図形のみが登録されている。一方、CMOS、BiCMOS、ECL等の各種テクノロジーのライブラリ設計を行う上で、テクノロジーによって大きく異なるのはリーフセル中の素子の部分であり、配線部分はその影響を受けない。

そこで、HSYLAではテクノロジーに依存するデータを含むリーフセルからセルシンボルを生成し、ブラックボックスとして扱うことにした。セルシンボルの入出力端子は、リーフセル中の各素子のコンタクト配置可能位置に対応する。リーフセルを

一つのシンボルとして扱うことにより、リーフセル設計以降の工程では、セルシンボルとコンタクト、配線シンボル等を使用してテクノロジーに独立なシンボリックレイアウト設計が可能となる。

2-3. 階層設計

ゲートアレイの需要が拡大するにつれて多種多様なライブラリが必要になってきており、大規模なライブラリも要求されてきている。しかし、レイアウトライブラリが大規模になるほど、その設計に時間を要し、既に作成済みのライブラリを組み合わせ、新規のライブラリを作成する階層設計が重要となる。

そこで、H S Y L A では作成済みのライブラリを、リーフセルと同様にセルシンボルとして扱い、それらを組み合わせ階層設計を行えるようにした。作成済みライブラリ中の配線データは、セルシンボルと分離されて配線シンボルに変換されるため、必要に応じて下位階層の配線を修正することも可能である。

2-4. 回路シミュレータとの結合

レイアウトライブラリの設計は、デザインルールを満たしているだけでなく回路の性能を保証する必要がある。つまり、ライブラリの設計期間短縮には、レイアウト設計の効率化だけでなく、回路シミュレータと有機的に結合し、レイアウトライブラリ設計工程におけるフィードバックを最小に抑えることが重要である。

H S Y L A ではシンボリックレベルの結線情報から回路シミュレータ用のネットリストを生成し、レイアウトに変換する以前にライブラリの機能検証を行うことができる。またレイアウト生成後は配線長に従った寄生容量も含めてネットリストを生成することにより、配線遅延を考慮した詳細な

回路シミュレーションが行える。

2-5. シンボリック配線データの再配置

2-1節で、デザインルールに独立なレイアウト設計手法の必要性について述べた。しかし、H S Y L A はリーフセルをセルシンボルとして扱うため、デザインルールが更新された場合は新たに作成したリーフセルからセルシンボルを生成し直さなければならない。そこで、H S Y L A には、作成済みライブラリのシンボリック配線データを、更新後のリーフセルシンボルに合わせて再配置する機能を持たせた。再配置されたシンボリック配線データを、新規デザインルールに従ってレイアウトデータに変換することにより、ライブラリのデザインルールを容易に更新できる。

3. システム構成

図1にH S Y L A システムの構成を示す。以下、システム構成を設計フローに従って説明する。

- ①. リーフセルを用意する。この際、トランジスタレベルのシンボリックレイアウトシステム S Y L A ^[1] を使用して設計を行えば、デザインルールの異なるリーフセルを容易に作成できる。
- ②. テキストエディタでパラメータファイル、及び配置ファイルを作成する。パラメータファイルには、作成ライブラリセル名、使用リーフセル名等を記述する。配置ファイルには、リーフセルの配置位置を言語記述する。
- ③. シンボリック入力画面ジェネレータは、パラメータファイルと配置ファイルを入力し、リーフセルからセルシンボルを生成する。その後、配置ファイルに従ってセルシンボルを配置し、シンボリック入力画面を生成する。
- ④. グラフィックエディタ上でライブラリ

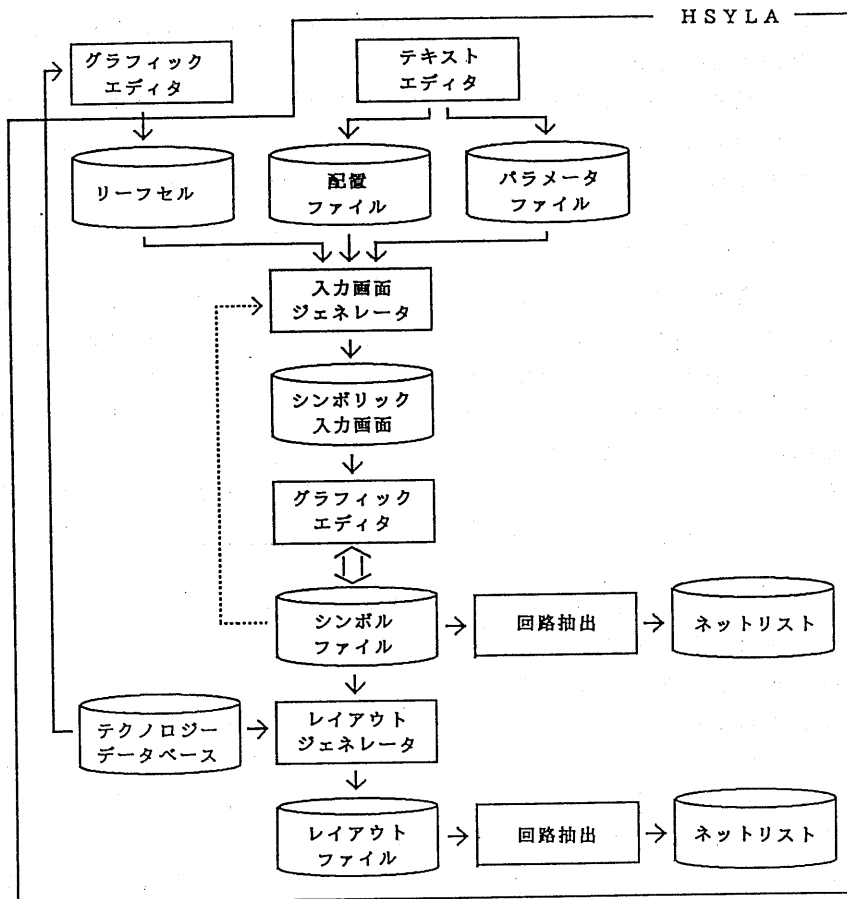


図1 システム構成

を作成する。ここでは、配線間のトポロジーだけを考慮してシンボリック配線する。

- ⑤. 回路抽出プログラムは、シンボルファイルに含まれるシンボル間の接続情報からトランジスタレベルのネットリストを抽出する。このネットリストは回路シミュレータに入力され、回路の機能検証が行なわれる。
- ⑥. レイアウトジェネレータは、シンボルファイルからリーフセルと同一のデザインルールでスペーシング処理したレイアウトデータを生成する。
- ⑦. 生成したレイアウトデータから、再度回路抽出プログラムによって配線の寄生

容量が含まれたネットリストを抽出し、配線遅延を考慮した詳細な回路シミュレーションを行う。

- ⑧. 階層設計を行なう場合、既に作成したシンボルファイルを一つのリーフセルとみなし、②から⑥と同様の手順でライブラリ設計を行なう。
- ⑨. ライブラリのデザインルールを更新する場合、まず新規デザインルールのリーフセルを用意する。シンボリック入力画面ジェネレータは、パラメータファイル、配置ファイル、及び、旧デザインルールのシンボルファイルを入力し、シンボリック配線データを新規デザインルールのリーフセルシンボルに従って再配置する。

4. HSYLAによるライブラリ設計

本章では、階層シンボリックレイアウト手法を使用したライブラリ設計について、具体例を挙げながら説明する。

4-1. シンボリック入力画面の生成

リーフセルレイアウト中にはコンタクトを配置可能な位置に入出力端子を設定するか、コンタクトを直接配置する。シンボリック入力画面ジェネレータはパラメータファイルで指示されたリーフセルを入力し、リーフセルの図形データをゲートアレイの下地を構成するマスク層の図形と、ライブラリ配線層を構成するマスク層の図形とに分離し、前者からセルシンボルを生成する。後者は、シンボリック配線データの一部と

してライブラリ配線層に登録される。その際、リーフセル内にコンタクトがあった場合は、セルシンボル内の同位置に入出力端子が発生され、上位配線層との接続情報が保持される。シンボリック入力画面ジェネレータは、ライブラリ設計に使用する全てのリーフセルシンボルを生成した後、配置ファイルに記述された情報に従ってそれらを配置し、シンボリック入力用の画面を作成する。図2に例を示す。

既に設計済みのライブラリを組み合わせで大規模なハードマクロを作成する場合、個々のライブラリをリーフセルとみなしてセルシンボルを生成する。その場合、ライブラリ内の配線図形は全て上位配線層のシンボリック配線データに変換される。

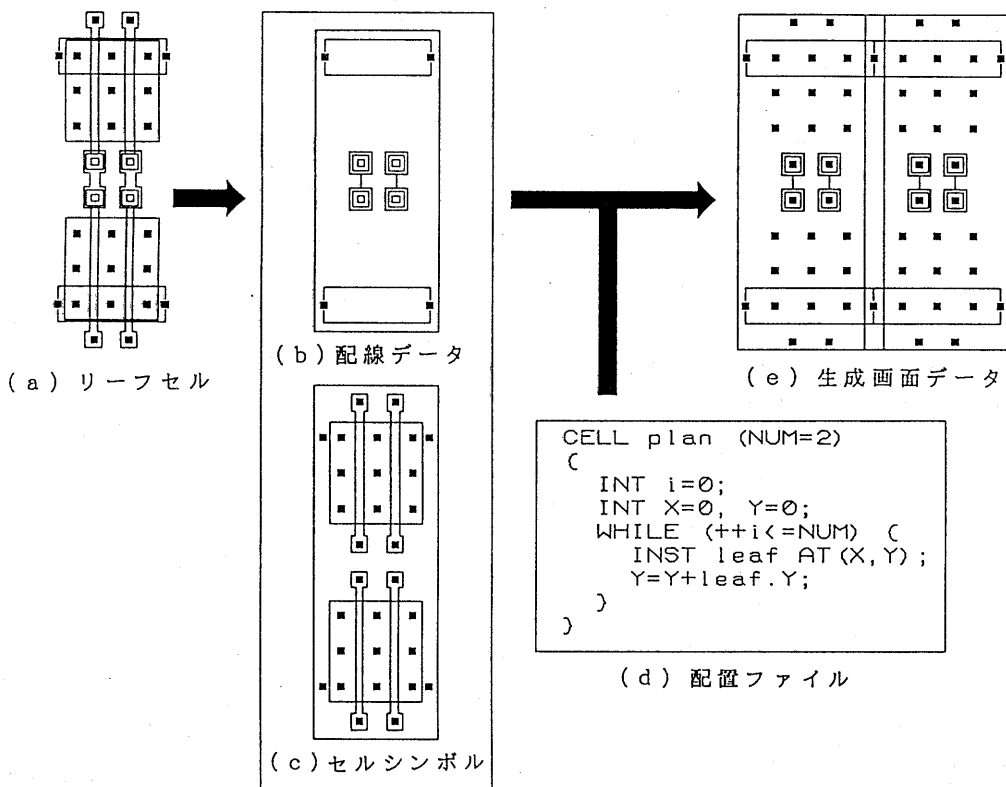


図2 シンボリック入力画面の生成

4-2. シンボリックライブラリ設計

ライブラリ設計は、配線、コンタクト、スルーホール、外部ターミナルの各シンボルを用いてセルシンボル中に置かれた入出力端子間をシンボリックに配線する。この際、コンタクトシンボルはリーフセルシンボルの入出力端子上にのみ配置できる。

エディタは、リーフセルシンボルの外形と入出力端子のみ表示するモードと、リーフセルシンボル中の素子も含めて表示するモードを切り換えられるようになっており、シンボリック配線の際の、設計者の便宜を図っている。図3に2種類の表示モードを示す。

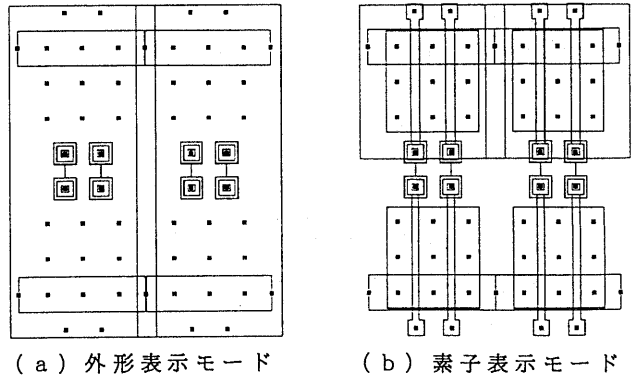
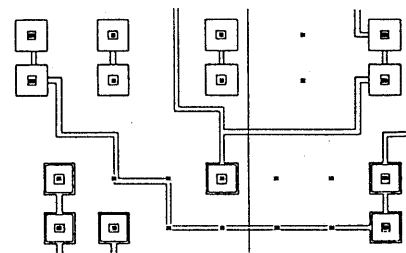


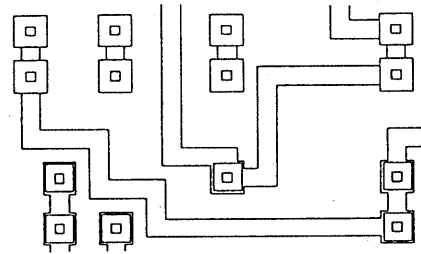
図3 セルシンボル表示モード

4-3. レイアウトデータの生成

レイアウトジェネレータはシンボリック設計されたライブラリ配線層から各シンボルとそのネット情報を抽出した後、テクノロジーデータベースに定義された配線層のデザインルールに従ってスペーシング処理を施し、シンボリックデータを実レイアウトに変換する。スペーシング処理ではコンタクト及び外部ターミナルシンボルは下地マスク層との位置関係を保存するため配置座標が固定される。即ち配線シンボルの経路とスルーホールシンボルの位置のみがデザインルールを満たすよう左、及び下方向に局所的に変更される。図4に例を示す。



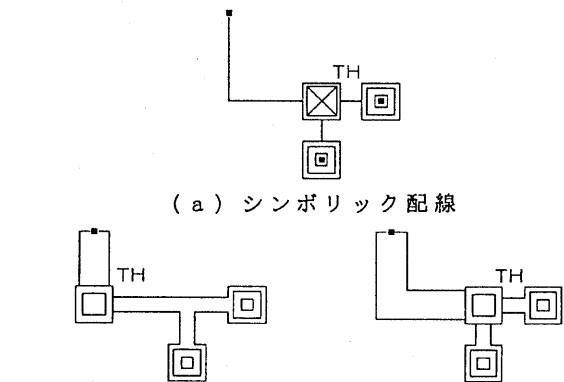
(a) シンボリック配線結果



(b) 生成レイアウト

図4 レイアウトデータの生成

図5 (a) のようなケースで、スルーホールシンボルの位置を変更したくない場合には、ANCHORと呼ばれる特殊なシンボルをスルーホールシンボルと重ねて置くことにより、その位置を固定することができる。図5 (b) がANCHORシンボル未使用の場合、図5 (c) が使用した場合の生成レイアウトである。



(b) ANCHOR未使用 (c) ANCHOR使用

図5 生成レイアウトの制御

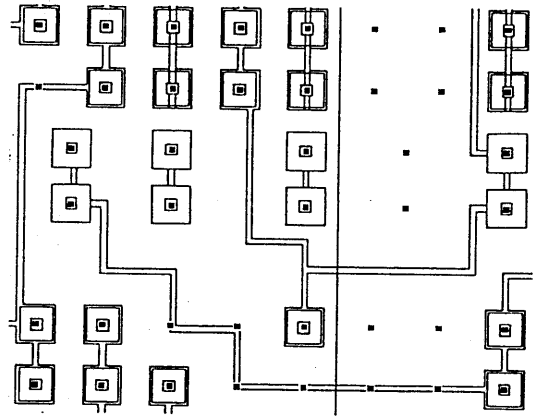
4-4. デザインルールの更新

作成済みライブラリのデザインルールを更新する場合は、まずリーフセルを再設計した後、シンボリック配線データを新規デザインルールのリーフセルシンボルに従って再配置する。ライブラリ配線データを構成する各シンボルは、リーフセル中に設定された入出力端子位置を基準にして以下のようにその位置が決定される。

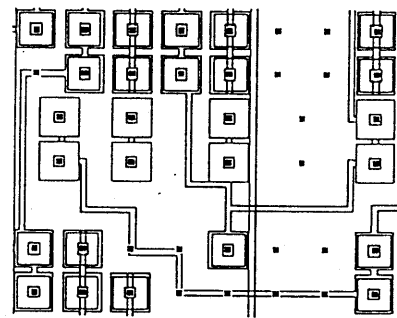
コンタクトシンボルはシンボリック設計の際必ずセルシンボル中の特定の入出力端子に置かれる為、リーフセル更新後も対応する端子位置にその配置座標が移動される。

スルーホール及び外部ターミナルシンボルの配置座標と配線シンボルの経路座標は、セルシンボル中の入出力端子との相対的な位置関係を基に更新後の座標位置が算出される。

本システムでは、配線シンボルに対して始終点での接続シンボル情報と配線の接続方向がプロパティとして与えられている。従って、始終点シンボルの配置座標と配線経路座標のみを更新すれば配線シンボルはそのまま保存される。図6に例を示す。



(a) シンボリック配線データ



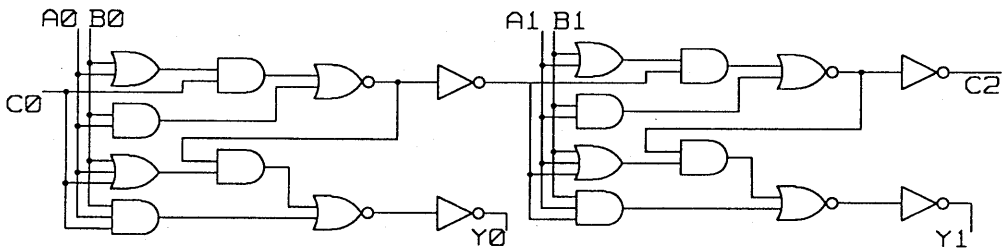
(b) 再配置されたシンボリック配線データ

図6 デザインルールの更新

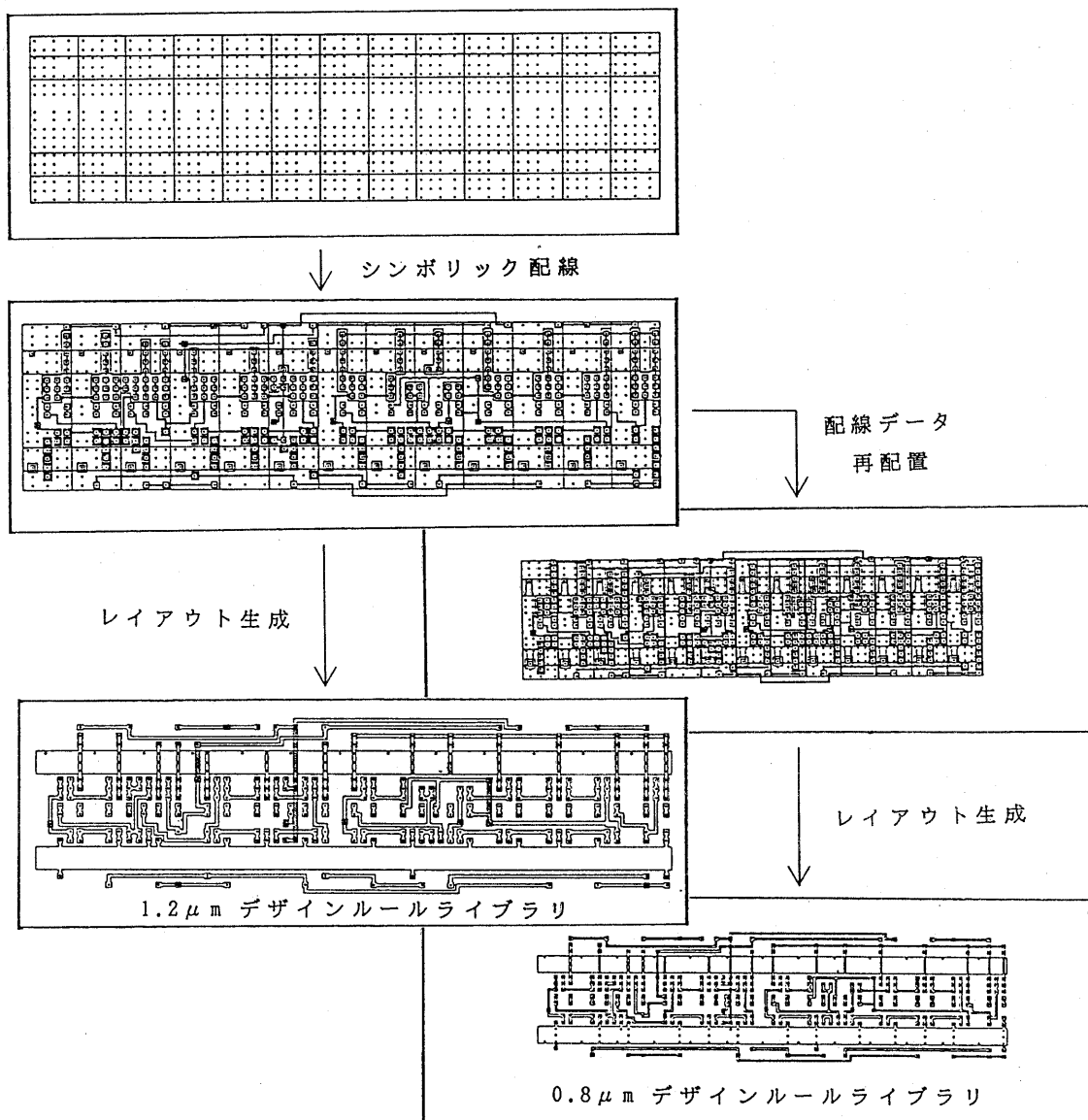
5. 適用例

図7にHSYLAによる2bit-adderの設計例を示す。シンボリック入力画面上で、セルシンボル中に置かれた入出力端子間をシンボリック配線することによ

りライブラリの設計を行ない、レイアウトデータを生成した。また、リーフセルのデザインルールを更新し、シンボリック配線データを再配置したシンボルファイルからレイアウトデータを生成した。



(a) 論理回路図



(b) レイアウト生成フロー

図7 2bit-adder設計例

6.まとめ

ゲートアレイのライブラリ設計を支援する階層シンボリックレイアウトシステムHSYLAについて述べた。本システムは、これまでにSSIからMSIクラスまでのライブラリやRAM、ROMなど、250種類のライブラリ設計に適用し、設計期間を従来の1/2に短縮した。また、ライ

ブライのデザインルールを更新し、新規レイアウトの生成が容易に行える事を確認した。

7.参考文献

[1] 植松 他：「シンボリックレイアウトシステムSYLAによるセル設計」
情報処理学会，設計自動化41-1