

VLSIレイアウトCADシステム“MCDA”

VLSI Layout CAD System “MCDA”

IC製造技術の飛躍的進展により、複雑なVLSIを効率よく設計する技術が重要となっている。本論文は、開発されたVLSIレイアウトCADシステムについて述べるが、特に従来問題であったチップのコスト競争力確保に対するシステムの解決策に焦点を合わせる。

設計効率の大幅向上に有効な自動レイアウトを中心に、最小チップサイズ実現のために、設計者のマクロな判断が可能な対話形CAD(ブロック分割・セル割付け、フロアプラン)を導入した。種々の設計試行を可能にするデータ管理を備え、一貫し、かつ柔軟なシステムとなっている。

5～40万トランジスタ規模のVLSIに適用し、従来の $\frac{1}{2} \sim \frac{1}{5}$ の工数で、人手設計に迫るチップサイズが実現できた。

山城 治*	Osamu Yamashiro
寺井秀一**	Hidekazu Terai
鈴木五郎***	Gorô Suzuki
高山 豊****	Yutaka Takayama
伊藤高明*****	Takaaki Itô
河野浩樹*****	Hiroki Kôno

1 緒 言

IC製造技術の飛躍的な進展により、ICに集積される回路規模は1970年代の数千トランジスタのレベルから1980年代には数十万トランジスタのレベルに達し、より複雑化してきた。1981年に発表されたHP(ヒューレットパッカード)社による45万トランジスタ規模の32ビットマイクロプロセッサは、正に超LSI時代の幕あけであった。しかしそれは、複雑なLSIを製造はできても、設計能力が追従できないという切実な問題を提起したものであった。従来の人手に頼っていた設計では、製品仕様の決定から最終的なLSIチップを得るまでに、数十人の設計者を投入しても数年はかかると予想されている。

マイクロコンピュータ用LSI設計の分野では、マイクロプログラム論理あるいは階層化されたモジュール設計など、複雑な設計をなるべく容易に設計できるような手法が導入されてきており、成功している。このような設計手法の確立による設計効率向上は非常に重要ではあるが、完全な解決策ではなく、特にASIC(Application Specific IC)のように、応用分野が広くて特定することができず、標準化なども困難なものでは、設計手法だけによる対応は不可能と言える。

複雑化する超LSI設計への完全な解決策は、CAD(Computer Aided Design)システムの構築であり、それは今まで設計者の行なっていためんどろで、時間がかかり、誤りを犯しやすい仕事を肩代わりできるものでなければならない。日立製作所は、これに答えるVLSIレイアウトCADシステムMCDA(Microcomputer Chip Design Automation System)を開発した。本システムでは、自動配置配線を含むレイアウト設計支援を中心にしているが、論理設計からマスクアートワーク設計まで、すべての設計工程を一貫して支援している。

従来、自動配置配線により設計されたLSIは、人手設計されたものに対し、チップサイズが大きく、コスト競争力がないとされていたが、本システムでは、設計者の試行を支援するメカニズムを新たに導入し、人手設計に迫る最小チップサイズが得られている。今までに5～40万トランジスタ規模のLSI・VLSI設計に適用され、セル設計を含めて従来の $\frac{1}{2} \sim \frac{1}{5}$ の設計期間、工数低減の実績を挙げている。

本論文では、開発されたVLSIレイアウトCADシステムMCDAについて述べるが、特にチップのコスト競争力確保に対するシステムの解決策に焦点を合わせて説明する。

2 VLSI設計の流れとシステム基本思想

図1は、半導体メーカー内で通常とられる設計の流れを、製品仕様決定からアートワーク及びマスク製作工程までを示している。重要なポイントは、セル設計が論理設計と並行して進められることである。もちろん、セルがあらかじめ設計されデバッグ済みの場合もあるが、次のような場合にはセル

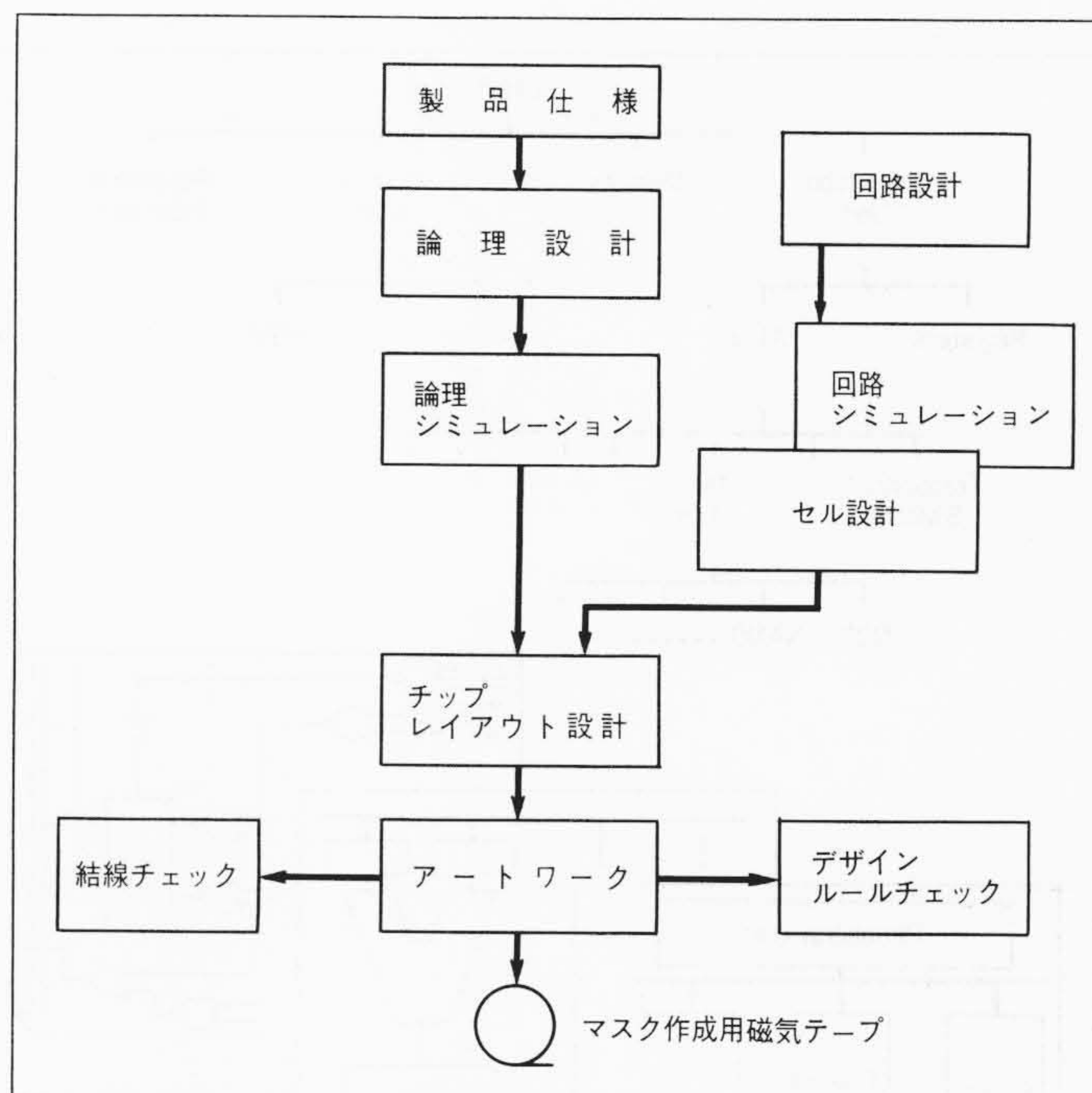


図1 VLSIの設計工程 製品仕様が決定されると、論理設計と並行してセルの回路設計とパターン設計が進められる。

* 日立製作所武蔵工場 ** 日立製作所中央研究所 工学博士 *** 日立製作所日立研究所 ***** 日立製作所茂原工場
***** 日立製作所高崎工場 ***** 日立マイクロコンピュータエンジニアリング株式会社

設計自体が重要となる。

- (1) 新しい製造プロセスあるいは回路を適用した製品系列を設計する。
- (2) 応用分野ごとに特別な仕様の回路(高速演算回路、アナログ回路など)が必要である。

ところが、今まで発表されているLSIレイアウトCADシステムは、いわゆるセルベースのシステムであり、設計しようとするLSIの論理回路をセル単位で記述するものとなっている²⁾。したがって、このように論理設計に応じて最適なセル設計をしなくてはならない半導体メーカー内での設計環境は、ユーザーである設計者に次のような難しい制約を課すことになる。つまり、LSIチップ上に実現しようとする回路はすべてあらかじめセルとして設計し、ライブラリに登録しておかなければ、論理設計、論理シミュレーションの着手すらもできないことである。

- 以上の認識からVLSIレイアウトCADシステムとして、
- (1) 論理設計とセル設計が互いに独立して進められるように、論理設計と実装設計の間の橋渡しを行なう手段をもつことが必要であり、更にコスト競争力をもつために、
 - (2) 設計者の最適チップ設計に向けての試行データの管理機構の導入
- が必要であり、それを実現することをねらった。

トップダウン設計あるいはボトムアップ設計いずれの設計手法が採られようとも、LSI設計工程は、論理設計と実装設計の二つの主工程に分けられる。論理設計工程での設計者の関心事は、LSIの論理的な機能の実現である。それに対し、実装設計工程での関心事は、具体的なLSIチップの実現であり、電気的特性の確保のための回路、製造技術の選択決定、コスト最小へのチップサイズ最小化が重要課題である。

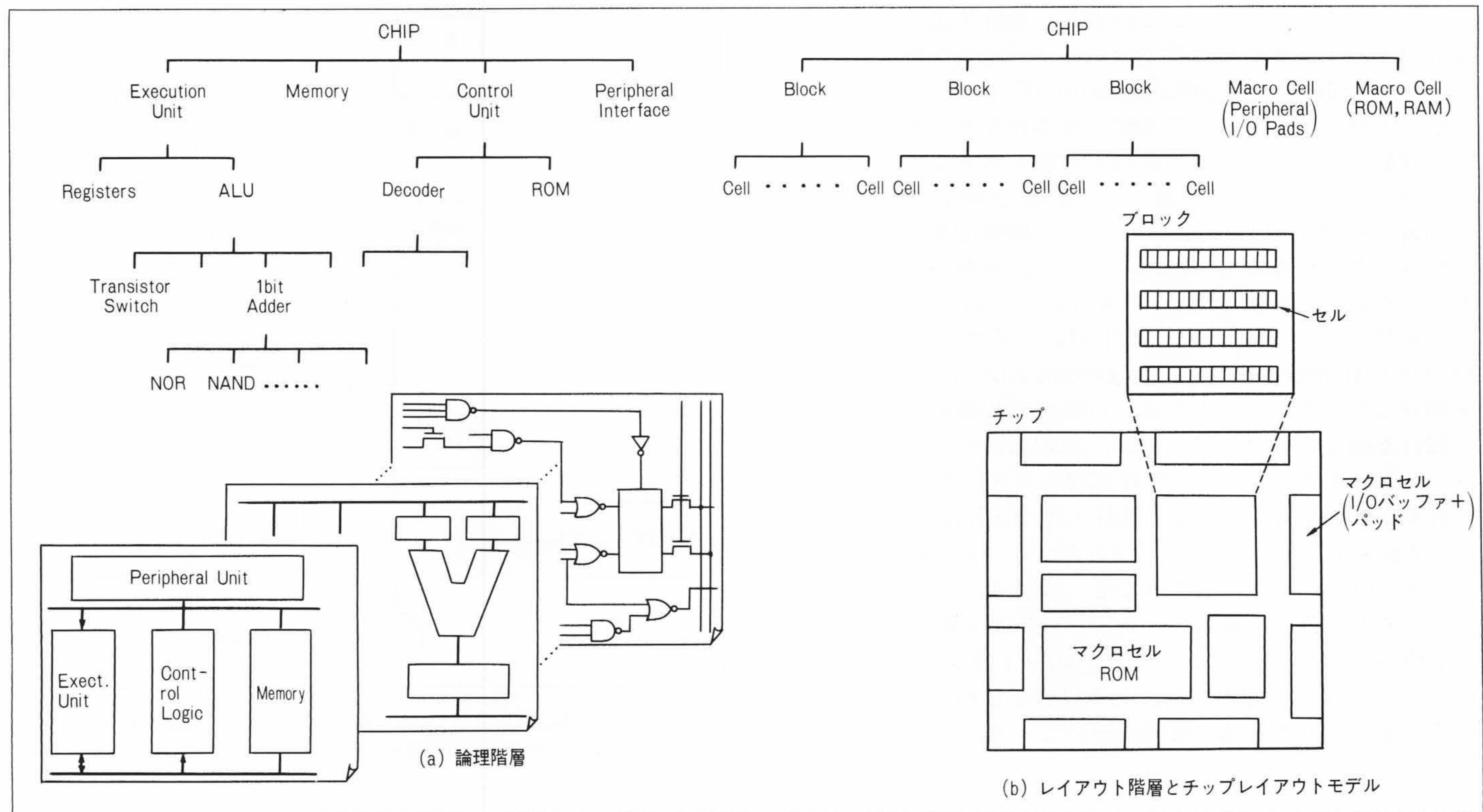
したがって、図2に示すようにLSI設計結果の表現方法は、論理設計と実装設計(=レイアウト設計)で異なり、各々のデータの階層構造、内容はかなり違ったものとなっている。論理設計データは、NAND、NORといった論理ゲートを単位に表現され、階層の深さはLSIの論理機能表現するのに必要なだけとられる。それに対し実装設計であるレイアウトデータは、セルを配置配線した結果として表現され、その階層は自動レイアウトプログラムの取扱い能力により制限されることが多い。

本システムでは、図2(b)に示すように、レイアウトデータは二つの階層をもっている。つまり、LSIチップはブロックあるいはマクロセルから構成され、ブロック自体はセルを配置配線した結果として構成されている。マクロセルには、ROM(Read Only Memory)、RAM(Random Access Memory)、I/O(Input/Output)バッファとボンディングパッドを含む周辺ブロック、あるいは演算回路など、特性的に重要で特に人手設計されたブロックが当てはまる。

VLSIの最適設計に向かって設計者がなんら制約を受けることなく本システムを使ってゆけるように、論理からレイアウトへのデータを、設計者が対話的に変換できるツールを導入した。それにより設計者は次のことが可能となる。

- (1) 論理設計結果での階層構造を、ブロックの展開あるいは新しいブロックへの組換えなどにより、レイアウト設計に適した階層構造に再構成する。
- (2) 回路やセルの設計選択をレイアウト設計に入る時点まで保留しておき、論理設計の自由度を増す。

これにより途中での設計変更、特別仕様の回路の集積、あるいはその後の技術変遷にも容易に対応できる、非常に柔軟なシステムに仕上げられている。



注：略語説明 ROM(Read Only Memory), RAM(Random Access Memory), I/O(Input/Output)

図2 VLSIにおける論理設計表現とレイアウト設計表現 論理階層では機能ごとの階層をもつが、レイアウト階層では大きさを意識した階層をもつ。

3 システム構成と各機能

図3は本システムの構成を示したものである。論理設計データはグラフィック端末を対話的に用い、回路図面の形で入力する³⁾。入力された論理データは、双方向のMOS (Metal Oxide Semiconductor) トランジスタ、論理ゲート及び論理ブロックを統一的に扱える論理シミュレータにより検証される⁴⁾。特性的にクリティカルな部分は、回路シミュレーションにより、電圧、電流のレベルでも詳細に信号遅延などが検証される。

論理データ変換ツールにより、設計者はレイアウトブロックへの分割と、使用するセルの割付けを対話的に行なう。対話形フロアプランは、設計者に早い段階でLSIチップの概略イメージを与えることができ、最小チップサイズに導いていく。グラフィック端末から設計者の入力したブロックの概略配置を基に、フロアプランは各ブロック間の配線領域を推定し、最終的に配線可能なブロック配置位置を決定する。自動配置配線プログラムは、ブロック内の自動配置配線⁵⁾と、フロアプランにより決定されたブロック配置に基づいたブロック間自動配線に分かれている。

自動配置配線結果は、マスクレベルの実体パターンであるアートワークデータに変換され、デザインルールチェック⁶⁾及び結線チェックにより最終データとしての検証が行なわれる。結線チェックを備えているのは、特性上、チップサイズ最小化などでアートワークデータへの最終の変更調整が設計者により行なわれるためである。

データ管理は、上記の設計過程での設計データの変遷(抽象的な論理データから具体的なアートワークデータへの進化)を記録し、途中途中での設計データのバージョン及びライブラリのコントロールを行ない、種々の設計試行から最適な設計

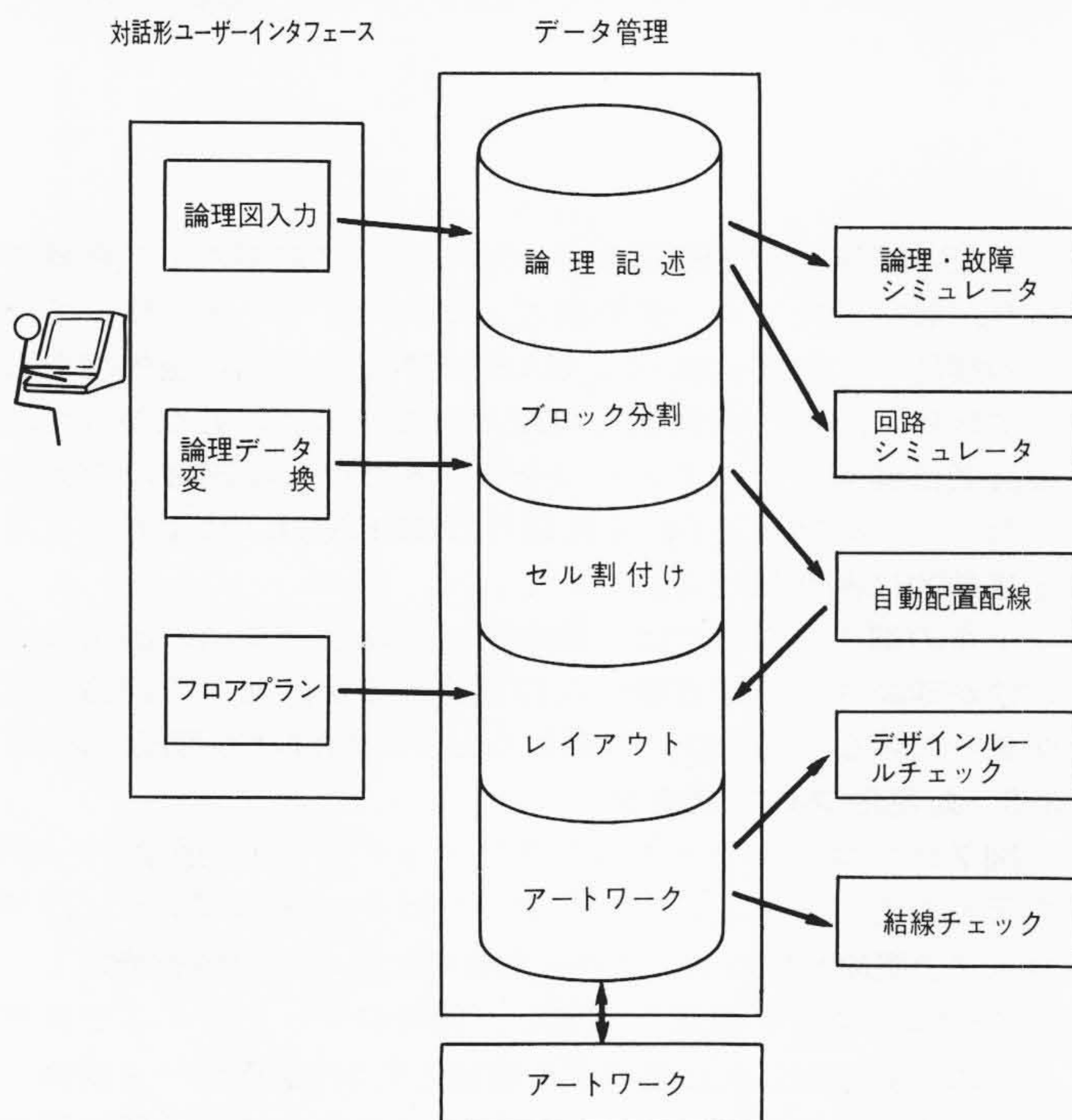


図3 VLSIレイアウトCADシステム“MCDA”の構成 自動配置配線を中心に、論理入力からマスク作成用磁気テープまで一貫したシステムとなっている。チップサイズの最小化のために、設計者の判断が入られる対話形のユーザーインタフェースをもっており、データ管理機構により誤りのない設計試行を支援している。

データを選択してゆくことを可能にしている。

本システムの特徴は、単に自動配置配線プログラムだけの開発でなく、論理設計からマスクデータ作成までを設計エラーが入らないように一貫して支援していること、及び各々の設計過程で対話的なユーザーインタフェースを通して、設計者の介入を許すことにより極めて柔軟なシステムになっていることである。特徴的である対話入力部分について次にやや詳しく説明する。

3.1 論理図入力システム

この論理図入力システムは、TSS (Time Sharing System) のグラフィック端末を用いたもので、世の中に出ているワークステーションを用いたスタンドアロンタイプと異なり、HITAC Mシリーズの大形計算機上に管理される設計データを直接入力編集できることが大きな特徴である。5万ベクトル/秒の高速図形表示能力により、TSSでありながらスタンドアロンタイプ並みの対話性能をもっている。図4は、本システムにより入力された論理図面の一例を示したものである。本システムでは、半導体メーカーとしての設計ノウハウに基づき、種々の工夫が取り入れられているが、次に挙げるのがその一例である。

- (1) 階層化された図面の入力だけでなく、フラットな大図面の入力が、図面サイズ、素子の規模に実用上制限されることなく、可能である。
- (2) 図面中で繰り返し使用される回路ブロックを、図5のように図面内にマクロ図として定義し、使用できる。

3.2 論理データ変換

本ツールは二つのユーザーインタフェースをもっている。一つはブロック分割であり、もう一つはセル割付けである。

ブロック分割には図5に示すような四つの基本コマンドがある。

(1) GROUP

そこに含まれる素子を指定し、新しいブロックを切り出す。

(2) EXPAND

ブロックを内部回路に展開する。

(3) SPLIT

一つのブロックを二つの新しいブロックへ分割する。

(4) JOIN

複数の指定されたブロックを一つの新しいブロックにする。

設計者は上記のコマンドを組み合わせ、論理設計データのブロック構造をレイアウト的に最適なブロックに変更できる。しかし、設計者が変更できるのは階層構造だけであり、結線関係は全く変更することはできないため、これによるエラーが入り込む余地はなく、安心して種々のブロック分割を試みることが可能である。

セル割付けでは割付け自体のほかに、割り付けられたセルの機能が、論理的に一致しているかどうかのチェックもプログラムが行なう。セル割付けには図6に示すような三つの基本コマンドがある。

(1) DEFAULT

何も指定がないときに標準的に割り付けるセルの一覧表を作成する。

(2) TASSIGN

論理ゲートのタイプ別にセルを割り付ける。

(3) CASSIGN

個別の素子ごとにセルを割り付ける。

ここでタイプとは、NAND、NORなど論理機能を定義したものであり、個別素子とはそれを使用した一つ一つのゲート

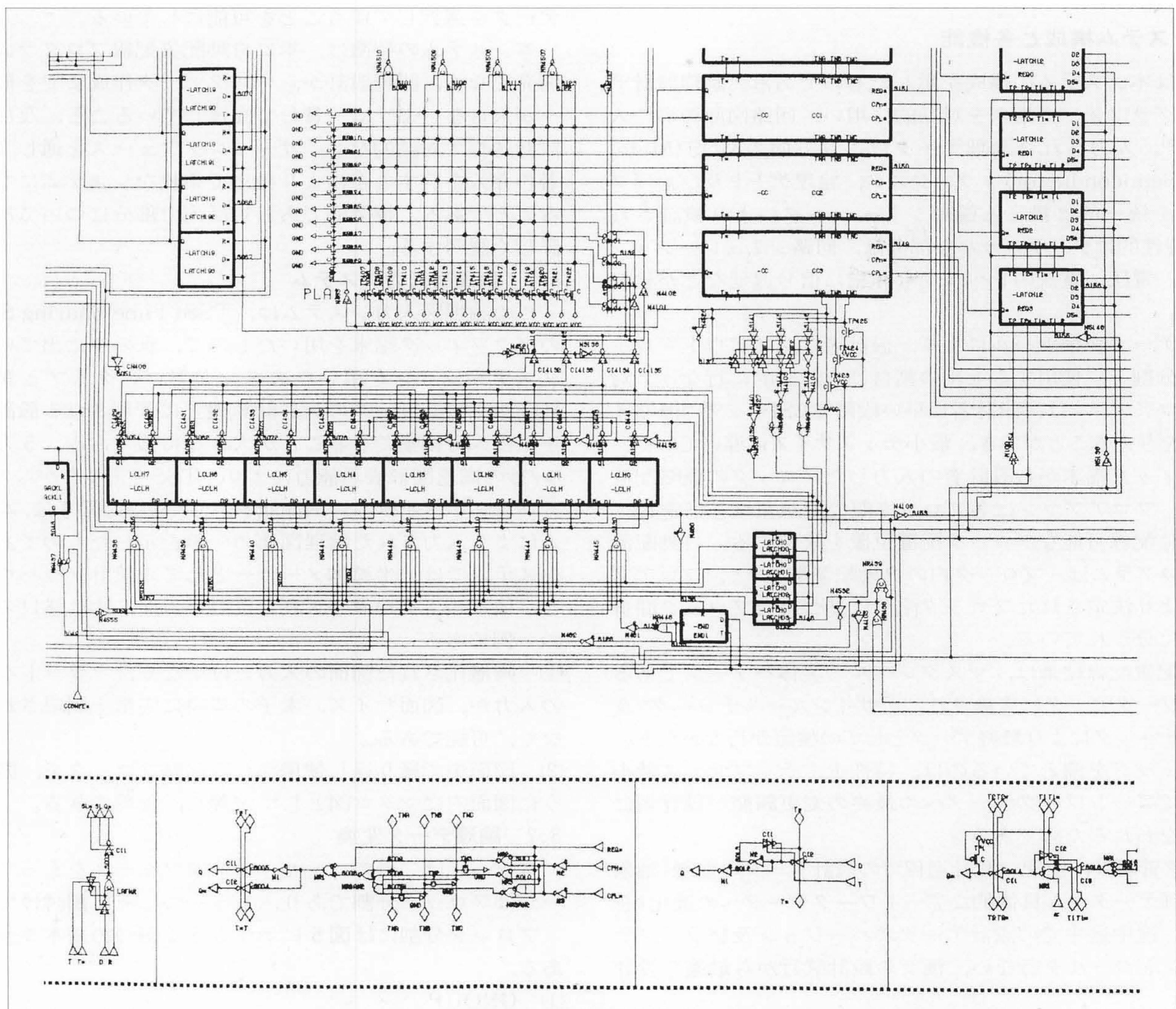


図4 論理図入力システムにより入力された論理図 VLSIの論理図面の取り扱いに便利のように、フラットな大図面が入力でき、図面中には繰り返し使われる回路ブロックをマクロ図面として定義できる。

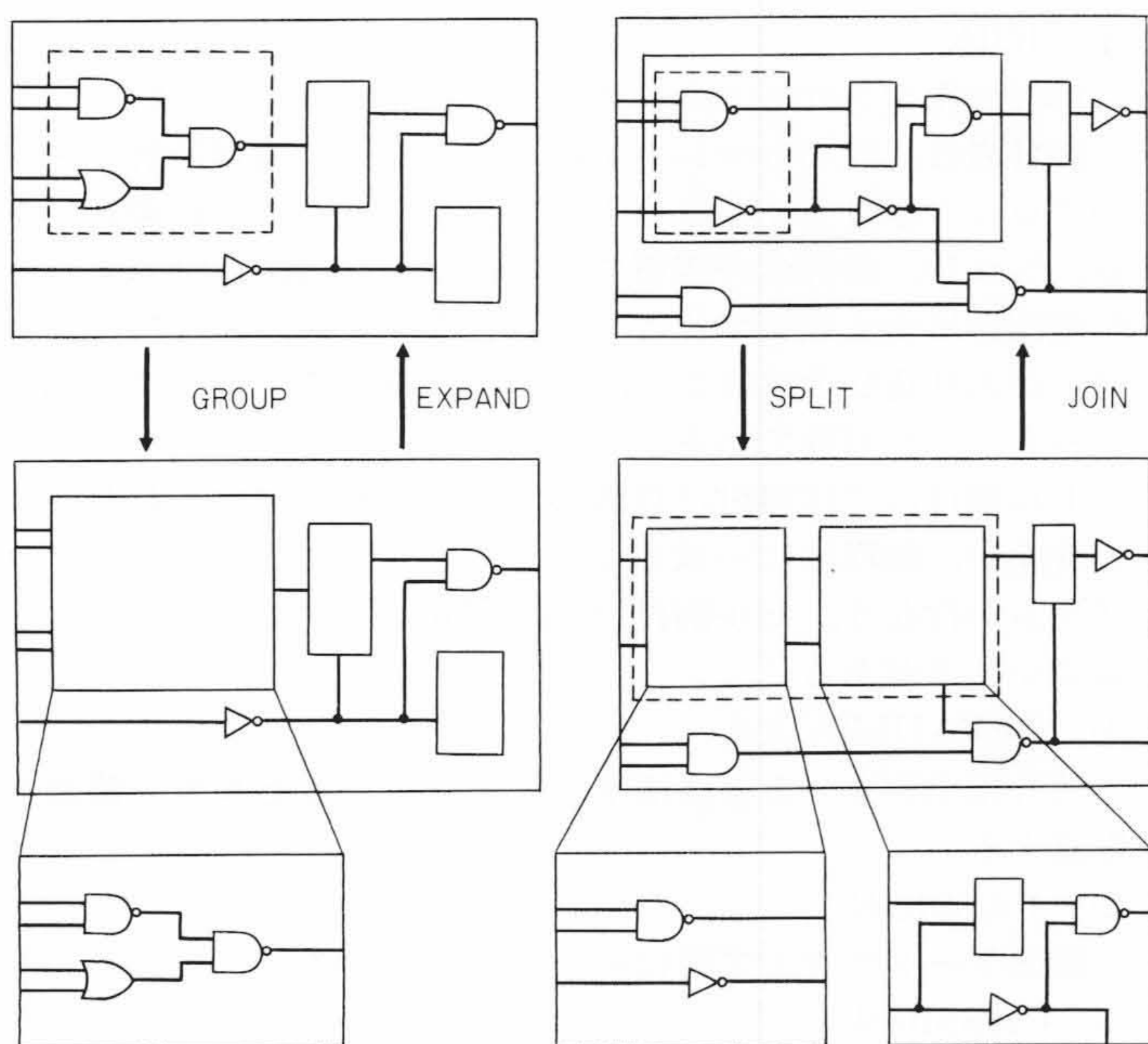


図5 論理データ変換でのブロック分割基本コマンド 設計者は、GROUP、EXPAND、SPLIT及びJOINの四つのコマンドを組み合わせ、レイアウトに最適なブロック階層構造に変換する。

を言っている。

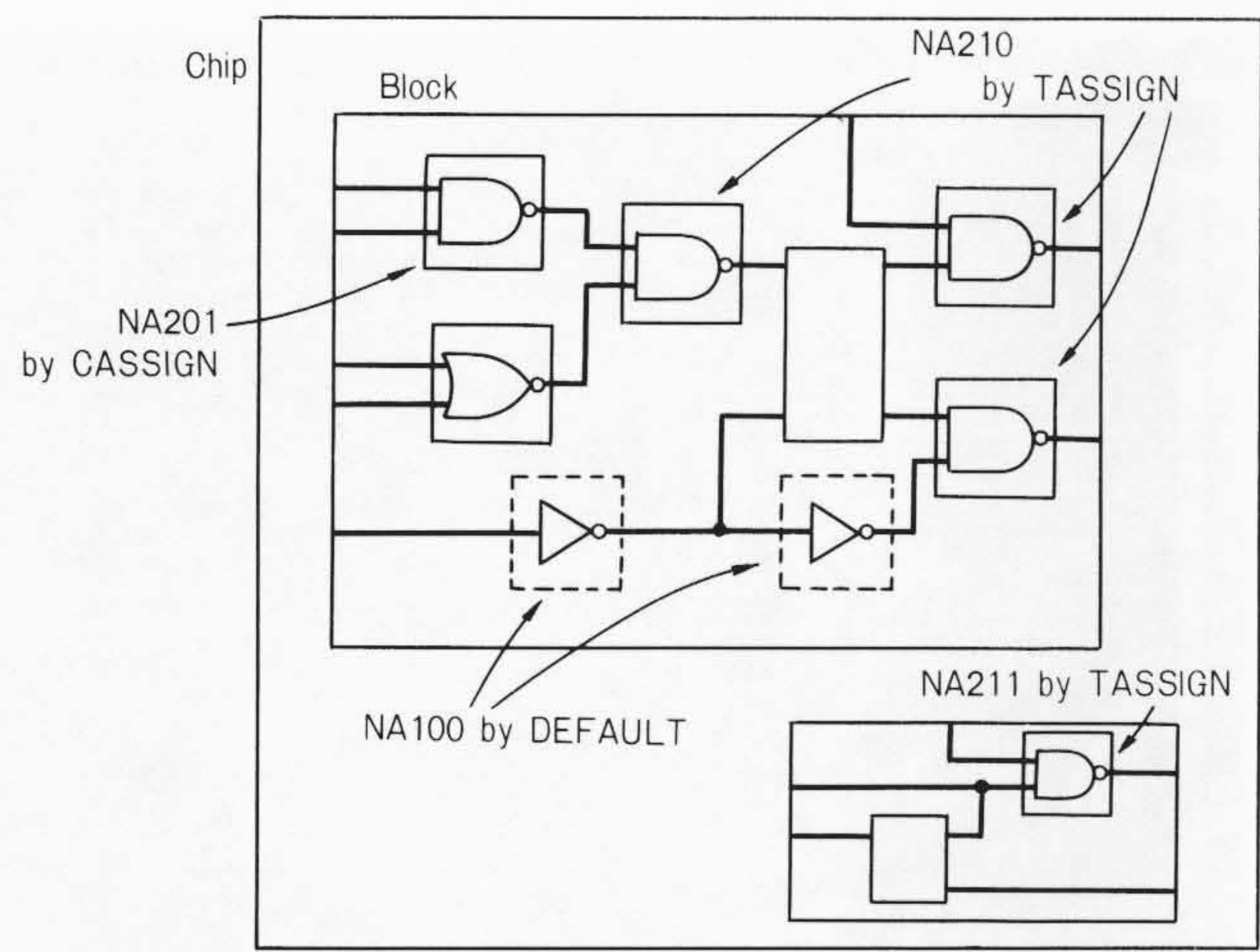
TASSIGNとCASSIGNは一つのブロック定義の中で有効であり、設計者はブロック定義ごとにこのコマンドを使ってセルの割付けを行なっていく。CASSIGNが最も高い優先度を持っており、次にTASSIGN、DEFAULTと続く。したがって、設計者は個別のクリティカルなゲートだけセル割付けの指定を行なっていくとよく、それ以外はDEFAULTによりほとんど自動的に割り付けられていく。

セルの端子の割付けは、論理側の回路とセル側の回路の一致性が確認されれば自動的に行なわれるが、セルの回路記述が整っていない場合は、対話的なピンの割付けも可能である。

3.3 対話形フロアプラン

図7はフロアプランでのグラフィック端末上の画面の一例を示したものである。フロアプランは先に述べたように、ブロックの概略配置から、ブロック間の配線必要領域を推定し、最終的なブロック位置を決定し、同時にチップサイズを見積るものであるが、次に説明する機能と3.2の論理データ変換ツールとあいまって、自動レイアウトシステムでの長年の目標であった人手設計並みのチップサイズ実現が極めて効果的に達成できている。

まず、フロアプランとブロック内自動配置配線が有機的に結合され、フロアプランのコマンドとし、ブロックごとの種々



標準セル割付けデフォルト表

Logic Type	STD Cell
INVERTER	NA100
2input NAND	NA200
2input NOR	NR200
.	.
.	.
.	.
.	.

図6 セル割付けコマンドとその方法 設計者は同じ論理ゲートでも、電気特性に従って最適なセルを割り付けていく。特にクリティカルでないゲートには、標準セルが自動的に割り付けられる。

の形状(=縦横比)をあらかじめ作っておき、ブロックの概略配置時に、チップ全体にむだなスペースが生じないような最適形状が選択できる。

また、ブロックの配置についても、どのブロックを近くに配置したらよいか即座に分かるように、ブロック間の結合度あるいは配線チャンネルのヒストグラム等々、設計者判断に必要な情報を提供できるようになっている。

フロアプランにより、各ブロックの大きさに極端過ぎるほどのアンバランスなどがあることが分かった場合には、ブロック分割に戻って変更することにより、更に最適化を図ることも可能である。

4 設計試行を支援するデータ管理

今まで説明したように、本システムでは設計者の介入を極

めて容易にしておき、設計者はなるべく多くの設計試行データをもととする。設計データは論理データに始まり、ブロック分割、セル割付け、フロアプランなどを経て、抽象的なデータからLSIチップを実現するためのより具体的な設計データが、本システムとの対話により追加されてゆくことができる。

本システムでのデータベースは、したがって各設計フェーズで設計者が準備作成した設計データの個別ファイルの集合として管理されている。設計フェーズとは設計工程の中で、比較的独立して進めることができる設計の作業単位である。日立製作所はVLSIの設計工程を次の設計フェーズに区分した。

- (1) 論理設計
- (2) ブロック分割

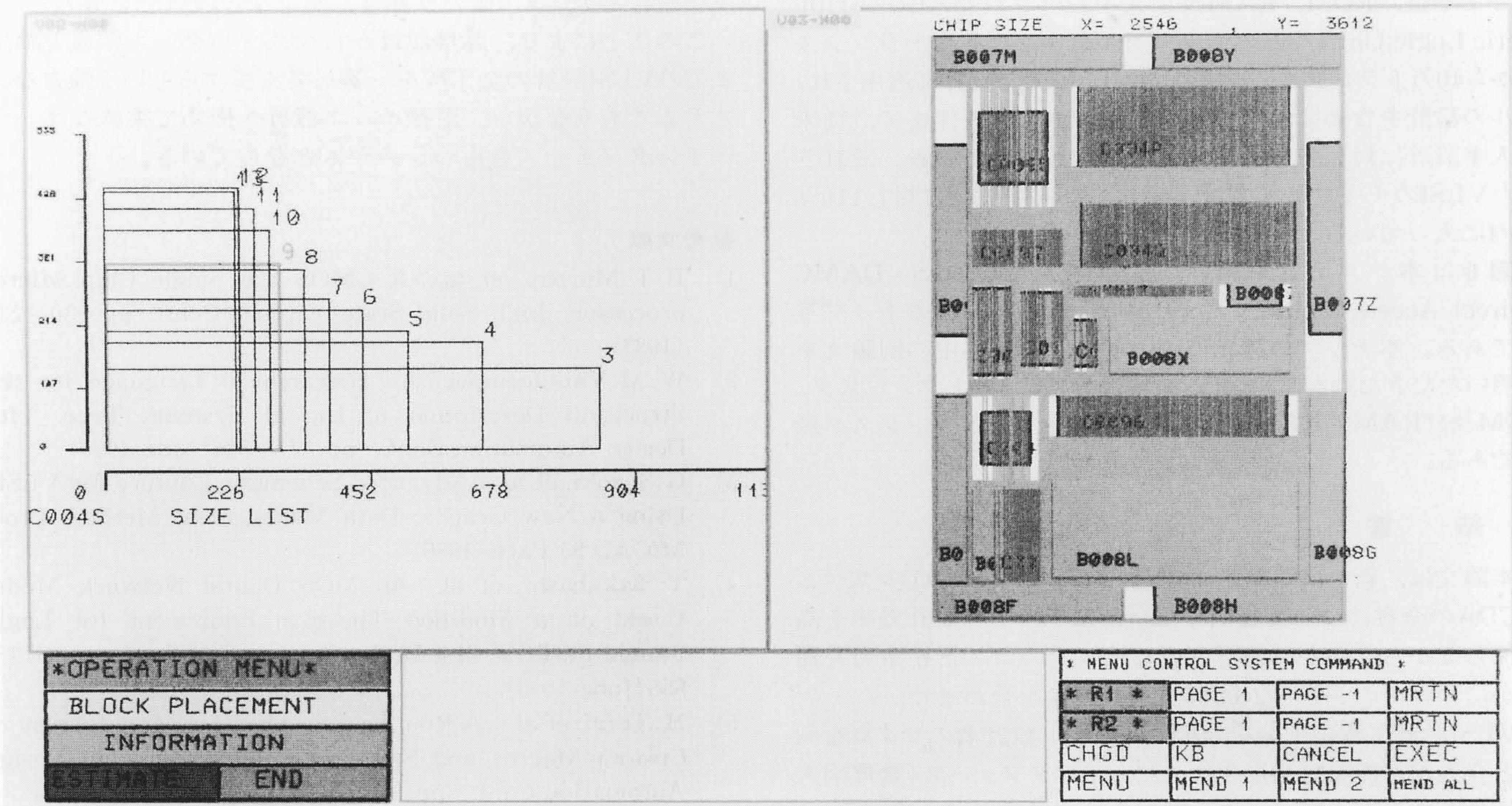


図7 フロアプランの例 フロアプランとの対話により、設計者は最適なブロック形状、ブロック配置及び最終的なチップサイズが簡単に分かる。種々なフロアプランを試みるにより、最小のチップサイズが得られる。

- (3) セル割付け
- (4) フロアプラン
- (5) アートワーク

各設計フェーズに対し、設計者が準備あるいは作成する入力データがある。その入力データには、設計試行のケースの違いや設計変更を示す異なるバージョンのデータが存在する。例えば、ある論理設計データに対し、ブロック分割の異なったデータが作られるであろうし、またそのブロック分割の一つに対し、異なったセル割付け方法があると考えられる。

フロアプランでもチップサイズ最小化に対し異なったブロック形状とブロック配置が考えられ、アートワークの段階でも最後の特性合わせ込みとして配線パターンの変更なども考えられる。

データ管理は、各設計フェーズでの設計データのバージョンの親と子の関係を記録しておく。また同時に、設計ステータスとして作業中か、完了かのいずれかを管理し、下位の設計工程で作業中に不用意に上位工程で設計変更が行なわれ、互いに矛盾した設計データの関係になることを防いでいる。

この機構により、アートワークデータを結線チェックにより検証するとき、容易かつ確実に比較すべき基となる論理設計データを見付け出すことができ、本システムの設計信頼性を高めている。

設計フェーズ、設計バージョン及び設計ステータスという極めて簡単なデータ管理の概念により、設計者は本システムを真に自分のツールとして容易にコントロールすることができる。設計者が知っていなければならないことは、正に次の3点である。

- (1) 今、自分が何をしているのか(=設計フェーズ)。
- (2) どのようなデータを入力として準備したか(=設計バージョン)。
- (3) 作業中か、完了したか(=設計ステータス)。

5 適用実績

今までに、ROM、RAM及び人手設計されたALU(Arithmetic Logic Unit)などのマクロセルを含んだ5万トランジスタから40万トランジスタ規模のVLSIチップの設計に適用され、セルの設計を含めて論理設計完了からマスク製作まで、従来の人手設計に対して $\frac{1}{2} \sim \frac{1}{5}$ の工数で設計できている。設計されたVLSIのチップサイズは、おおむね人手設計に対し110%以内に入っている。

図8は本システムを用いて設計されたCMOS・DAMC(Direct Access Memory Controller)のHD63450のチップ写真である。本チップの諸元については、本特集の別論文を参照いただきたいが、ランダム論理部分で6.8万トランジスタ、ROM及びRAMで1.2万トランジスタの計8万トランジスタ規模である。

6 結 言

本論文は、新たに開発されたVLSIレイアウトCADシステムMCDAの全容について説明した。特にその中で設計効率を高く保ちながら、いかに設計されたチップのコスト競争力を保つかにあつてのシステムの解決策に焦点を合わせた。

VLSI設計工程の十分な分析に基づき、設計者のマクロな判断と介入が可能な対話形のユーザーインタフェース(論理図入力編集システム、ブロック分割・セル割付け、フロアプラン)を導入した。また、種々の設計試行を可能にするデータ管理機構を組み込み、その中には設計フェーズ、設計バージョン

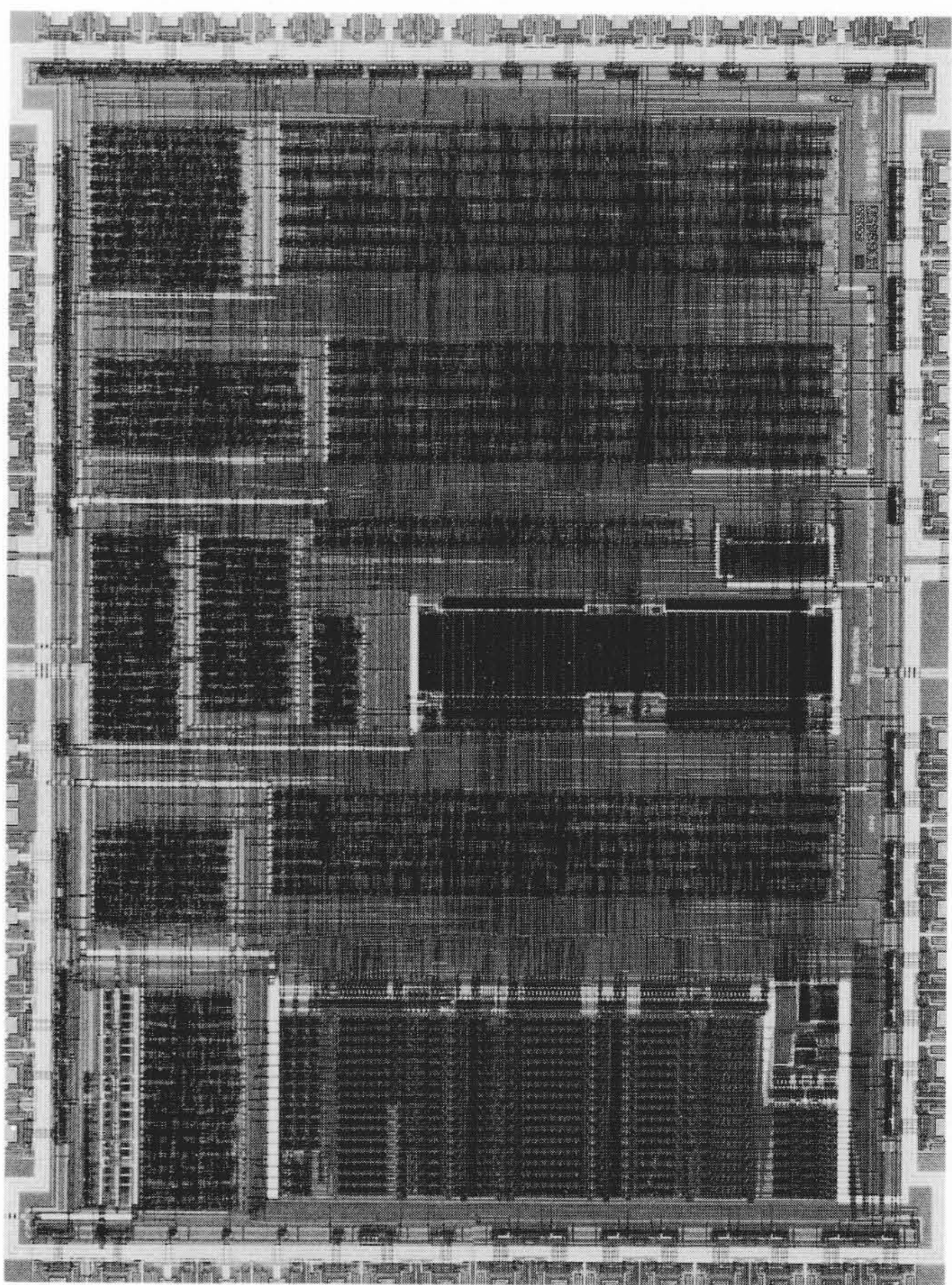


図8 本システムを用いて設計されたVLSIチップの例(CMOS・DAMC HD63450) ランダム論理6.8万トランジスタ、ROM及びRAM1.2万トランジスタ計8万トランジスタが集積されている。

及び設計ステータスという非常に簡単な管理概念を導入した。

このことにより、論理設計からマスクパターン作成と検証までのVLSI設計の全工程を一貫して支援するという強力なシステムでありながら、最適チップ設計へ極めて柔軟なユーザーインタフェースをもつシステムとなっている。

参考文献

- 1) B.T.Murphy, et al.: A CMOS 326 Single Chip Microprocessor, Int'l Solid-State Circuit Conf., pp.230~231 (1981)
- 2) W.M.VanCleemput: An Hierarchical Language for the Structural Description of Digital Systems, Proc. 14th Design Automation Conf., pp.377~385 (June 1977)
- 3) G.Suzuki, et al.: Advanced Schematic Capture for VLSIs Using a New Graphic Data Management Method, Proc. MICAD'85 Paris (1985)
- 4) T.Takahashi, et al.: An MOS Digital Network Model Based on a Modified Thevein Equivalent for Logic Simulator, Proc. 21st Design Automation Conf., pp.549~555 (June 1985)
- 5) H.Terai, et al.: A Routing Procedure for Mixed Array of Custom Macros and Standard Cells, Proc. 22nd Design Automation Conf., pp.503~508 (June 1985)
- 6) A.Tsukizoe, et al.: MACH: A High-Hitting Pattern Checker for VLSI Mask Data, Proc. 20th Design Automation Conf., pp.726~731 (June 1983)