

MOSマスク解析における論理検証

羽山 繁 , 渡里祐子 , 京井淳子 , 間野洋一郎
 (松下電器産業 (株) 中央研究所 半導体グループ)

1 はじめに

近年LSIは大規模・高集積となり、かつ高速・低消費電力を要求されている。そのため、一層の設計精度の向上と設計期間の短縮が急務とされている。

LSI設計におけるマスクレイアウト設計は、部分的に自動レイアウト処理を行いながらマニュアル設計を中心とするものと、マスタスライス方式のゲートアレイに代表される自動レイアウトによるものとに分けられる。いずれの場合も、マスクレイアウトの検証には、製造プロセスに関連する設計基準検査、電気的な特性検査等が必要であり、その的確なフィードバックが全体の設計精度、設計期間の短縮に大きなウェイトを占めている。

特に論理情報のミス、さらにレイアウトにより発生する寄生容量、抵抗等による動作スピード、消費電力、誤動作等の動作不良については今後とも十分な検証が必要不可欠である。

そのため、LSIのマスクパターンデータから回路レベルの情報を抽出するマスク解析システムが既にいくつか報告されている。1) 2) しかし、LSIの規模が増大するにつれ、処理時間・データ量の問題から、解析は困難となる。

これに対処するため我々は、MOS/LSIのマスクパターンデータを階層的LSI設計データベースで管理し、そのマスクデータを解析して、回路及び論理回路への変換、電気的エラーチェック、さらに回路/論理シミュレーション言語の自動生成まで含めた総合的なマスク解析システムPANAMAP-M/Lを開発した。3)

前回報告した、マスクパターンから回路変換を行うPANAMAP-Mに引き続き、今回は、PANAMAP-Mの結果を用いて、論理変換及び論理検証を行うPANAMAP-Lについて報告する。

本報告では、主に論理変換手法と論理図自動作成手法について述べ、本システムがMOS/LSI設計時の有効な論理検証手段であることを示す。

2 システム構成

PANAMAP-M/Lのシステム構成は、図1に示すように、前半のブロックPANAMAP-Mと後半のブロックPANAMAP-Lに大別される。

本システムでは、LSI・CADシステム等を用いて作成されたマスクパターン情報を、セル階層構造を保ったまま大型計算機上の設計データベースに格納し、処理対象データとしている。以下各ブロックを説明する。

PANAMAP-Mは、マスクパターン情報からMOSトランジスタの認識、接続関係の抽出、層間容量、回路定数の算出等を行い、回路情報を出力する。回路情報からは、トランジスタリスト、容量値リスト等の各リスト出力、回路図作成、回路シミュレータ：PANACAP用の入力言語作成等のアプリケーションを用いることができる。簡単な回路図をグラフィックで確認したり、XYプロッタで作図することも可能である。

さらに、回路変換結果のマスクパターン情報から、マスク解析結果の素子別デザインルールチェック、またはパターンの電気的レベルを考慮したデザインルールチェックを行うことが可能である。

PANAMAP-Lは、抽出した回路情報からトランジスタの相互接続によって論理ゲートを認識し、遅延時間等を算出して論理情報を出力する。論理情報からは、セル・ゲート間の接続リストの出力、論理図作成、論理シミュレータPANALOGIC用の入力言語作成等を行うことができる。論理図も回路図同様の出力処理が可能となっている。

これらの結果の回路情報と論理情報は、各々設計言語MSDLに自動変換して、マスクパターンデータとともに、半導体のための階層的設計データベース管理システムで管理しており、設計時のMSDLとマスクからのMSDL

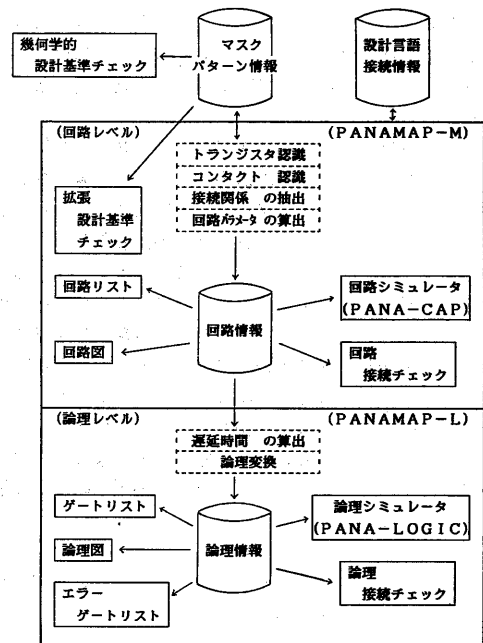


図1 システム構成図

との自動接続検査を可能としている。このような設計・検証・出図のための各アプリケーションシステム群は、データベースを中心にたがいに連結し、システムを構成している。

3 機能説明

3.1 階層的データベース

LSIマスクパターンの情報は、設計時のマスクパターンと1対1に対応した階層セル形式であり、図形・接続・階層情報で構成している。図形情報は、マスクパターンを、エリアとラインで各階層毎に独立して表現している。接続情報は、回路素子・論理素子間の相互接続を、ネット記述形式で表現している。階層情報は、図形の階層と接続の階層の両方を表現しており、セルの呼び出しと物理的配置情報、外部へ接続する端子の情報を持っている。端子は、端子名と位置座標を持ち、回路交換時にノード番号を付加することにより、端子・図形間の接続を表現することができる。このように、セル毎に情報を持っているため、重複データをもつことが避けられ、大規模データの処理が比較的容易となっている。

3.2 階層構造を利用した回路変換

PANAMAP-Mは、階層的データベースの図形情報（パターンデータ）と階層情報（内部セルデータ）から、LSIマスクの解析を行っている。

パターンデータについては、外部からカード形式で入力した層番号・プロセス等の情報を用いて、各層間で図形演算処理を行い、素子を認識し、素子間の接続を追跡して層間容量や回路定数等を算出している。また、内部セルについては、ピンと外側の図形との接続を認識し、階層間の接続を追跡することが可能なため、セルの内部はブラックボックスのままに解析を行うことができる。

解析結果は、設計言語MSDLとしてデータベースに格納され、さらに回路解析に接続したり、ネットリストや回路図を作成することができる。

3.3 論理変換

PANAMAP-Lは、PANAMAP-Mの結果の回路情報を用いて、各階層単位に論理変換を行っている。

3.3.1 論理ゲート認識方法

回路変換結果データであるトランジスタの接続関係から論理ゲートを認識する。内部セルについては、指定したセルのみ展開し、他のセルはピンのみを展開して、認識した論理ゲートと内部セルのピンとの接続情報を抽出することが可能である。NMOS、CMOSは構造上、認識手順に若干の違いがあるが、今回は、CMOSの場合の論理ゲート認識方法について、以下に示す。

- (1) 内部セルのピンを展開し、ノード番号を付け、論理情報ファイルに登録する。
- (2) 外部ピンを、論理情報ファイルに登録する。
- (3) トランスファーゲート、電気的エラーを有するトランジスタを見つけ、論理情報ファイルに登録する。
- (4) P型、N型トランジスタの各拡散部が、同一ノードのペアを見つける。
- (5) (4)で見つけたP型、N型トランジスタのペアの1つから、P型は電源まで、N型はグランドまでのパスを見つけ、パス上のトランジスタをすべて取り出す。
- (6) 取り出したトランジスタのP型群、N型群について、それぞれ直列接続数、並列接続数を求め、P型群の直列接続数、並列接続数が、N型群の並列接続数、直列接続数とそれぞれ等しい時は、通常ゲートの処理を行い、また、等しくない時は、トライステートバッファの処理を行う。
- (7) 通常ゲートは、N型トランジスタ群、P型トランジスタ群からそれぞれ論理を認識し、N型とP型で論理が逆になっていることを確認した後、論理情報ファイルに登録する。図2に論理ゲートの認識手順を示す。
- (8) トライステートバッファは、論理を確認した後、ドライブするインバータを探し、インバータを合わせて、論理情報ファイルに登録する。
- (9) (5)～(8)を、(4)で見つけたP型、N型トランジスタのペアすべてについて行う。
- (10) 登録されなかったトランジスタは、トランスファーゲートとして論理情報ファイルに登録する。
- (11) これまで認識した論理ゲートを用いて、ラッチやフリップフロップ等の複数の論理ゲートの組み合わせを認識し、マクロゲートとして論理情報ファイルに登録する。
- (12) 入出力ピンを自動発生する。













INVERTOR	HAND	NOR	TRANS
			
TR ISTATE	AND	OR	P-N Tr
			
D-LATCH			PF GATE
			

表1 論理ゲート一覧

以上のようにして認識したセル・ゲート間接続リストを図3に、認識可能な論理ゲートの一覧を表1に示す。

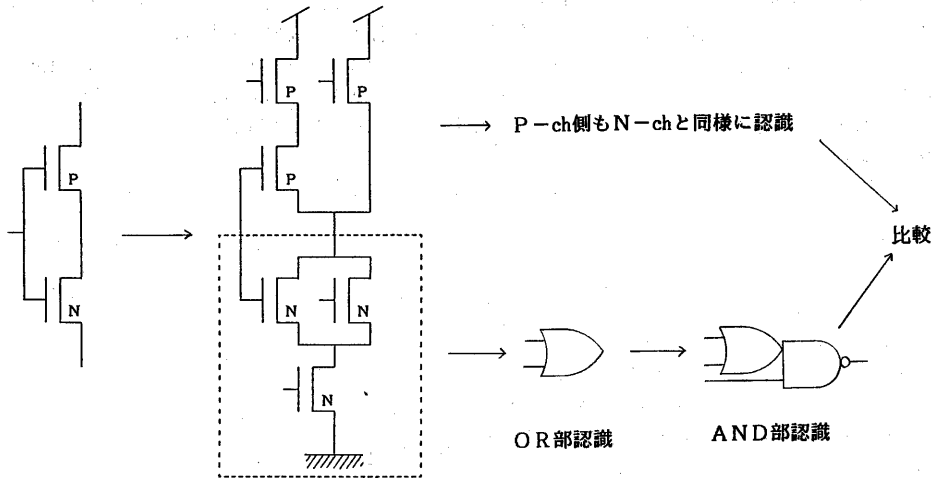


図2 論理ゲート認識手順

3.3.2 電気的エラーチェック
 3.3.1の1で述べたように、電気的にエラーと認められるトランジスタを、エラートランジスタとして抽出することができる。
 エラートランジスタの種類としては、以下に示す3種類が挙げられる。

*** LSAMP2\$ <U.000 > NESTING NODE LIST ***					
NODE. NO.	< PIN.NAME > < GATE. IO >	< CELL - ID > < GATE - ID >	< X(U) >	< Y(U) >	
N 000001	USS	AAANR1_000001	140	3	
	USS	EXNOR1_000001	70	228	
N 000002	UDD	AAANR1_000003	140	136	
	UDD	EXNOR1_000003	245	136	
	UDD	INJ011_000005	31	86	
N 000003	ID1	AAANR1_000001	178	0	
	ID1	EXNOR1_000003	238	0	
	O3	G.NAND_000006	255	-15	
	I1	G.INV_000012	86	155	
N 000004	ID1	AAANR1_000001	159	0	
	O1	G.NOR_000015	89	170	

図3 セル・ゲート間接続リスト

- (1) 電源・グランド接続エラー (2) ゲートノード接続エラー (3) ソース・ドレイン同一エラー

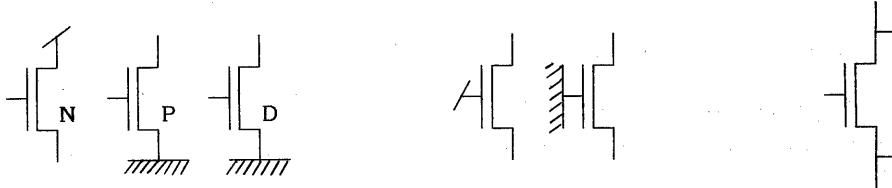
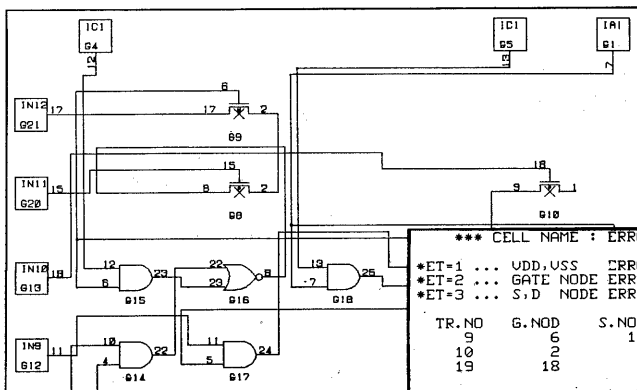


図4に、エラートランジスタの存在する場合の論理図、図5に、エラートランジスタリストを示す。



*** CELL NAME : ERRCHK\$ <U.001 > ERROR TRAN. LIST ***						
TR. NO	G. NOD	S. NOD	D. NOD	X(U)	Y(U)	ET
9	6	17	2	146.00	20.01	1
10	2	8	15	93.00	27.50	2
19	18	9	1	93.00	64.50	1

図4 エラートランジスタ論理図

図5 エラートランジスタリスト

3.4 論理図作成

論理変換の結果の論理情報ファイルより、論理図を出力し、設計者が図面上で論理を検証することができる。その際、検証しやすいように、マスク図面に対応した位置に論理シンボルを配置した論理図を出力している。

この論理図は、セルシンボル、ゲートシンボルを配置するゲート配置領域（メッシュ）と、メッシュ間を配線する配線領域とから構成される。1つのゲートは1つのメッシュとし、セルは、大きさ・形状に応じたメッシュ数とする。各々のメッシュの上下左右はすべて配線領域であるが、2メッシュ以上にまたがるセルがある場合、またがるメッシュの間は配線不可能領域とする。

以下に、マスク図面に対応した論理図の作成手順について、簡単に述べる。

< シンボルの配置 >

- (1) 論理変換後のセル・ゲート数の総和から必要なメッシュ数を決定する。
- (2) 入出力ピンをメッシュの周囲に配置する。
- (3) 各々のセル・ゲートを、マスク図面に対応した相対位置に一番近いメッシュに配置する。この時、複合ゲートは互に近い位置に配置する。
- (4) 見つけたメッシュに、すでに他のセルやゲートが配置されていた場合は、一番近い未配置のメッシュに配置する。

図6にシンボルの配置図を示す。

< シンボル間配線 >

すべてのセル、ゲートをメッシュに配置した後、下のメッシュから1行ずつ同一ノードのゲート入出力・内部セルの端子をつなぐように配線する。

- (1) 1行分のメッシュについて以下の処理を行う。
 - a. 1つのメッシュのセル・ゲートの入出力について、縦配線領域の空き位置まで配線を伸ばす。
 - b. 伸ばした配線のノードを、ノード別にカウントする。
 - c. そのノードの配線がまだ残っている場合は上に伸ばし、ない場合は、そこまでの縦配線を出力する。
 - d. 同一ノードの配線が隣り合うメッシュの間にある場合は、結んで結線記号を付け、一方のみを上へ伸ばす。
- (2) 1行分のメッシュの処理終了後、同一ノードの配線が2本以上あれば、次行のメッシュとの間の横配線領域を用いて結ぶ。さらにそのノードの配線がまだ残っている場合には、結んだ配線の中の1本のみを上へ伸ばし、他は結線記号を付けてそこまでの縦配線を出力する。

この時、次行とまたがるセルがあれば、その間は横配線禁止領域とする。

図7にメッシュ間、及び行間の配線手順を示す。

- (3) 全行について、(1)、(2)の処理を行う。
- (4) メッシュ及び配線の密度に応じて、自動的にコンパクションする。
- (5) 図形情報に変換してデータベースへ出力する。

さらに、入力ピンから出力ピンまでの信号伝達経路（パス）を重視した、論理としてわかりやすい図面を出力することも可能である。

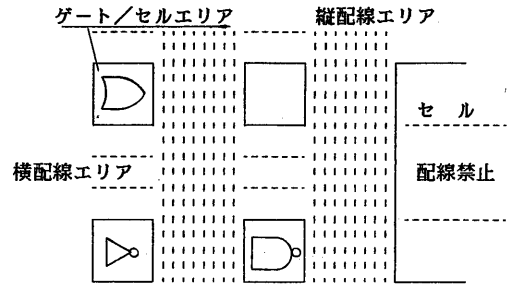
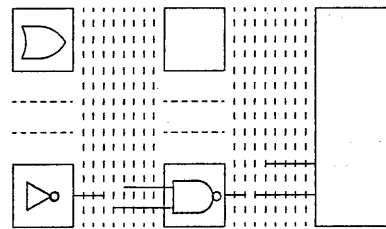
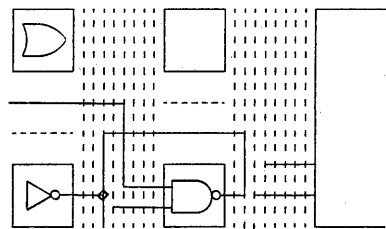


図6 シンボルの配置



(a) 縦配線エリアの割当て



(b) 横配線エリアの割当て

図7 配線手順

図8～10は、1セルのマスクパターン図、回路図、マスクパターンに対応した論理図であり、図11は、内部セルを持つ場合の論理図である。また、図12はバスを重視した場合の論理図を表している。さらに、多くのゲートを有する1ブロックの論理図を、一部分取り出して図13に示す。

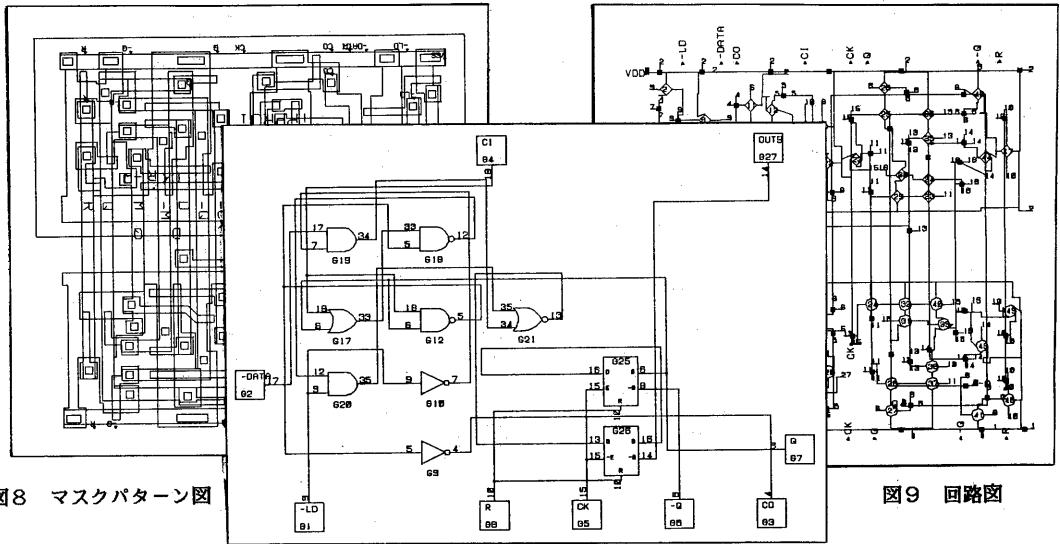


図8 マスクパターン図

図9 回路図

図10 論理図 (パターンに対応)

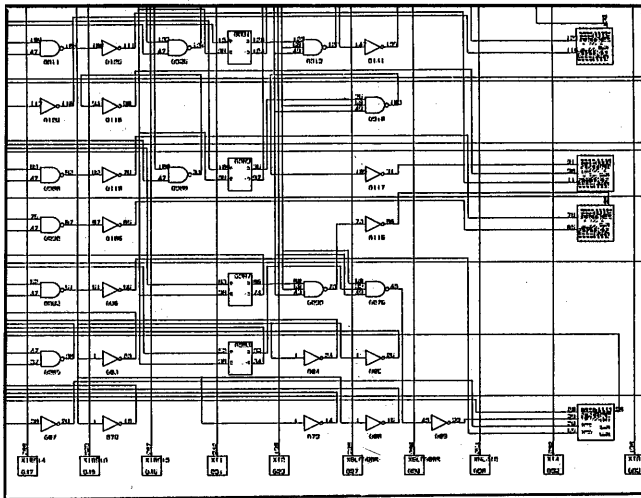


図11 論理図 (内部にセル有り)

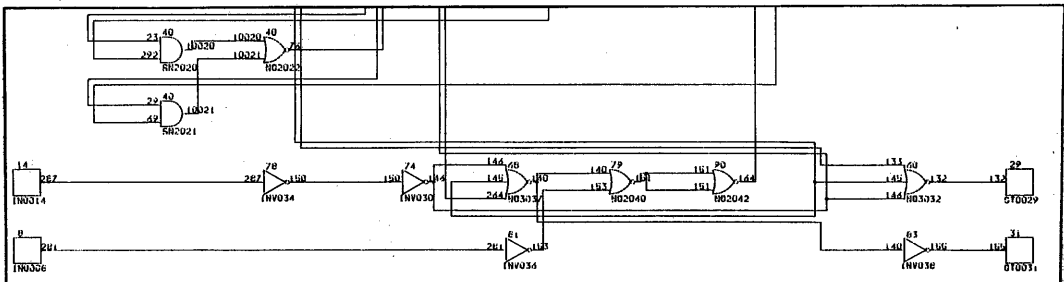


図12 論理図 (バスを重視)

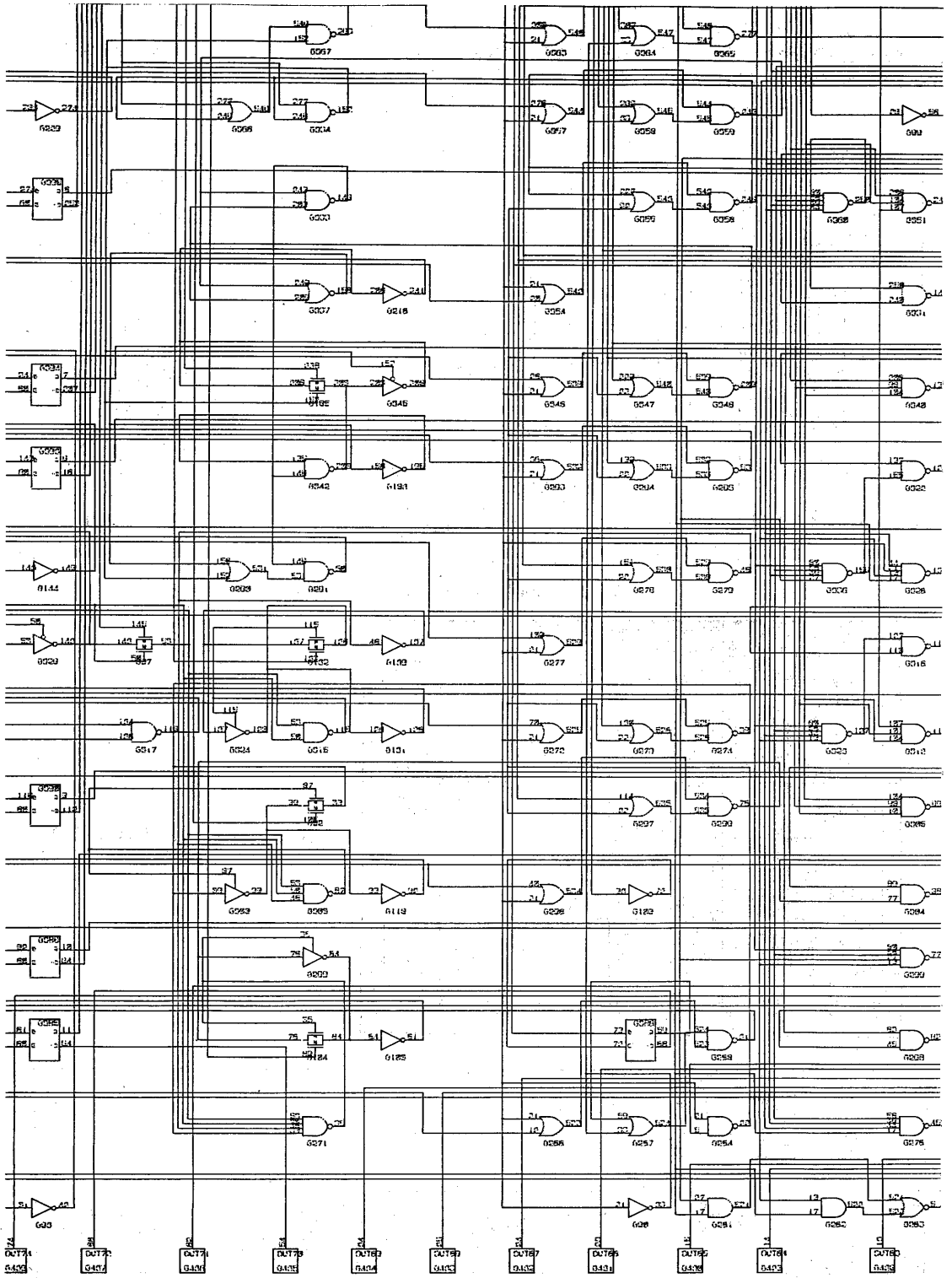


図 13 ブロック論理図 (パターンに対応)

3.5 論理シミュレータ用言語作成

論理変換の結果、ゲートを構成するトランジスタのチャンネルの長さや巾、ゲートの出力ノードにおける総容量値から、1ゲートによる遅延時間を簡易的に計算し、論理情報ファイルに登録する。その論理情報ファイルから、ゲートの接続情報、遅延時間情報を組み込んだ論理シミュレータPANALOGIC用の入力言語を自動作成することができる。

図14、15に、自動作成した論理シミュレータ用言語、及び論理シミュレータ結果を示す。

```

$$$$ PANALOGIC DATA $$$$$
$ MACRO DEFINE
$ NAND LATCH
IN1, IN2, OT1, OT2, TD1, TD2,
$ NATCH MACRO
OT1, NAND, TD1, IN1 OT2
OT2, NAND, TD2, IN2 OT1
$
$ ELEMENT
G0000010, INJ, 10, 7, -LD
G0000011, INJ, 8, 8, CK
G0000012, NAND, 12, 20, G0000013, CI
G0000013, NOR, 12, 17, H0000013, R
G0000014, TRFN, 19, 11, CK, G0000024
G0000017, OR, 0, 0, G0000013, CI
G0000018, NAND, 15, 12, G0000012, G0000017
$
$ PIN NAME
$ CO G0000029
$ O G0000013
$ -O H0000026
$ CUTS H0000027

```

図14 論理シミュレータ用言語

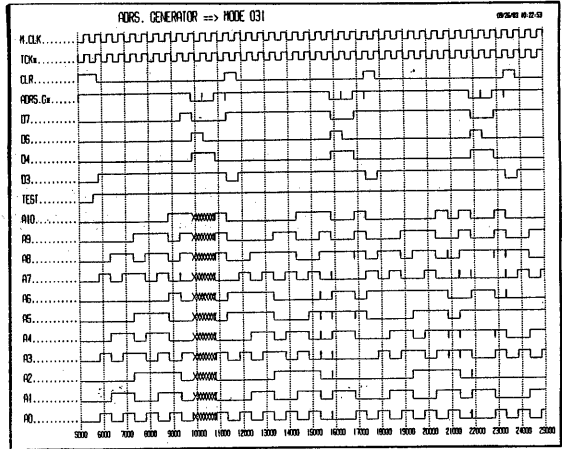


図15 論理シミュレータ結果

3.6 接続検証

論理変換の結果作成される論理情報ファイルを用いて、入力ピンから出力ピンまでのすべての信号伝達経路を見つけ出し、パスファイルを作成することができる。さらに、本システムでは入力ピンと出力ピンは全て固有の名称が付けられているため、各パスはそれぞれ独立の経路として認識できる。

パスファイルは、論理ゲートの位置情報に関係なく、論理的な接続のみを表現するものであるから、MSDLやPANALOGIC用言語からも作成することができる。従って、MSDLやPANALOGIC用言語から作成されるパスファイルと、マスク解析の結果作成されるパスファイルとを比較することによって、自動的に接続検証を行うことが可能である。

3.7 その他

3.7.1 拡張マスクパターン検証

マスクの図形情報は、回路変換時の図形論理演算によって、配線部、トランジスタ部、コンタクト部等の素子部に切り分られており、各々素子番号とノード番号が付けられている。そのため、各素子について、または同電位のパターンについて、個別にデザインルールチェックを行うことができる。たとえば、トランジスタ部の巾のチェックやコンタクト部と配線部との間のデザインルールチェック等である。また、さらに図形論理演算を用いることによって、その他の特殊なデザインルールも検証することが容易である。

3.7.2 拡張マスクパターン補正

先に述べたように、マスクの図形情報は各素子部に切り分られているため、ある素子部のみを、選択的に拡大または縮小することができる。また、素子間で図形論理演算を用いることによって、素子の追加及び変更が可能である。

4 おわりに

本システムは、階層的設計データベースを中心とし、MOS/LSIのマスクパターン情報から階層的にマスク解析を行うシステムに連結した、論理検証システムである。

MOS/LSIに対応するため、マスクパターン情報の階層構造によって処理対象データを削減し、処理時間の短縮をはかっている。さらに、論理設計の検証のため回路レベルから論理ゲートレベルに変換することで、出力データ量を削減し、設計時の情報と設計後のマスクパターンデータとの比較が容易にできる。また、階層構造を保ったままの論理変換と、さらにマスクの物理的配置情報を相対的に保ったままの論理図作成が行えるため、設計者の論理図面による検証も容易である。加えて、論理レベルで接続の検証が可能のため、接続検証の際に作成するパスファイルのデータ量も削減され、自動的な接続検証を短時間で行うことも可能になっている。

しかし、現時点では論理ゲートの認識手順がプログラムとして組み込まれているため、新しいタイプの論理ゲートを認識することが困難である。従って、認識手順をライブラリに登録できるようにする必要がある。また、抵抗値計算や、必要とするマスクパターンの一部を切り出した解析という要求があるため、今後はこれらの機能拡充を実現していく予定である。

5 謝辞

最後に、本システムの開発にあたり多大な御協力をいただいた、中央研究所 堀内副所長、石原グループマネージャはじめ半導体グループの方々、ならびに松下電子工業（株）CADグループの方々に深謝致します。

6 参考文献

- 1) 丹生，他：LSIマスクパターン情報，回路特性算出プログラム-MAICAP-，電子通信学会技術報告，SSD78-12，1978
- 2) S.Yamada ,etal：A Mask Patern Analysis System for LSI (PAS-1)，proc. 1979ISCAPS，pp858-861
- 3) 羽山，他：MOSマスク解析システム PANAMAP-M，情報処理学会 設計自動化 83-15-5