

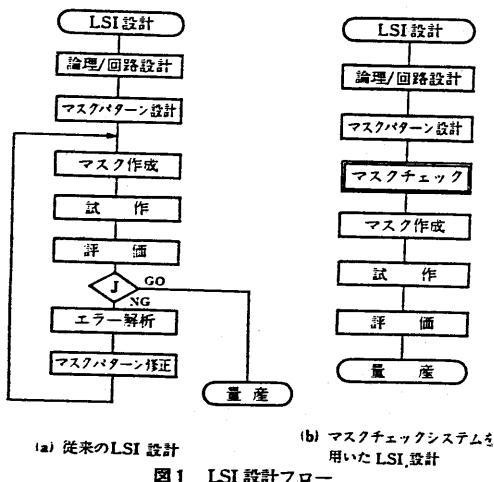
# LSI マスクパターン検証システム MIAS

小池 豊 安藤 宏 平川 和之  
(沖電気工業株式会社)

## 1. はじめに

近年のIC高集積化に伴い、LSIマスクパターン設計は、年々困難になつてゐる。短期間で誤りのないマスクパターンを作成するため、計算機による自動配置配線システムも活用されてはいるが、動作速度、消費電力およびチップ面積の最適化のため、限られたLSI以外ではかなりの人手設計介入が必要である。

マスクパターンデータは1,000ゲート程度の論理LSIでも数10万の開閉形で構成され、人間が完全にチェックすることは不可能に近い。マスクパターン設計の誤りは、試作テスト修正の繰り返しを招き、設計期間および設計費用を増大させる(図1-(a))。そのため、計算機によるマスクパターンの完全なチェックシステムの必要性は高い(図1-(b))。



筆者らは、すでに実用化し、高速であることが確認されているパターン幅、間隔チェックと同様のアルゴリズムを

用い、接続チェックを行うMIASシステムを開発した。チェックは、マスクパターンを電子回路/論理回路に逆変換することにより行う。逆変換された回路を回路シミュレータ/論理シミュレータに入力すれば、試作することなしにLSIの動作および性能が確認できる。MIASシステムは、すでに多数のLSIに対し運用され、種々のマスク設計の誤りを検出しており、その有効性が確認されている。

本論文では、MIASシステムの機能と構成および電子回路/論理回路抽出手法について述べる。

## 2. チェック項目と適用品種

1) 以下の4種類に分類されるすべてのチェックを行う。

i) 尺寸精度チェック  
ICプロセス等の制限からくる、パターン幅および間隔のチェック。

ii) 素子の完全性チェック  
素子部で、素子を形成するために必要な十分な图形の過不足チェック。

iii) 接続チェック  
要求を満たすために、適切な素子間を接続していることのチェック。

iv) 電気条件チェック  
要求性能を満たすために、各素子のパラメータが妥当であるとのチェック。

2) 適用品種を回路形式、プロセス、回路動作、規模に分類し述べる。

- i) CMOSおよびNMOS
- ii) シリコンゲートおよびアルミニゲートMOS
- iii) ロジック、メモリおよびアナログ回路
- iv) LSI-VLSI

### 3. システム構成

本システムのシステム構成図を図2に示す。各ステップでチェックされるチェック項目を表1に示す。

#### 3.1 マスクパターン

マスク設計装置（デジタルタイヤ、グラフィックディスプレイおよびプロッタ等で構成された装置）の出力であるマスクパターンデータを入力する。

#### 3.2 電子回路抽出

マスクパターンデータを素子形成部と配線形成部に分離する。素子形成部でトランジスタやコンデンサを認識し、配線形成部で、各素子間の結線を認識する。さらに素子や配線に寄生する寄生容量/抵抗を計算し、電子回路を抽出する（図3(a), (b)）。

#### 3.3 回路シミュレータ（回路解析）

抽出された電子回路は、回路シミュレータにより、電子回路レベルの動作確認を行う。回路シミュレータは、1ステップの大規模データは扱えないので通常小規模回路に分割して処理する。

#### 3.4 論理回路抽出

論理LSIの検証には電子回路レベルでは詳細すぎ、適さないために、電子回路を論理回路に変換する。変換はトランジスタの接続形態によりAND, NOR, NOT, OR, NOR, 等の論理ゲートに順次置き換えることにより行う。また、寄生容量/抵抗から、論理ゲート遅延時間に換算する（図3(c)）。

表1 各ステップでのチェック項目

適用	ステップ	チェック項目
論理LSI	寸法精度チェック	パターン最小幅。 パターン最小間隔(管内、管外)
	電子回路抽出	素子の完全性、浮き配線、電源線ショート、電源の供給、
	回路シミュレータ	ゲート数、ゲート幅が動作上妥当か、寄生容量抵抗値の妥当性。
	論理回路抽出	論理回路を構成するか、信号複数ショート、遅延時間が標準内か、
	論理シミュレータ	パターン設計による論理誤り、ダイレクションの妥当性。
	論理回路図出力	

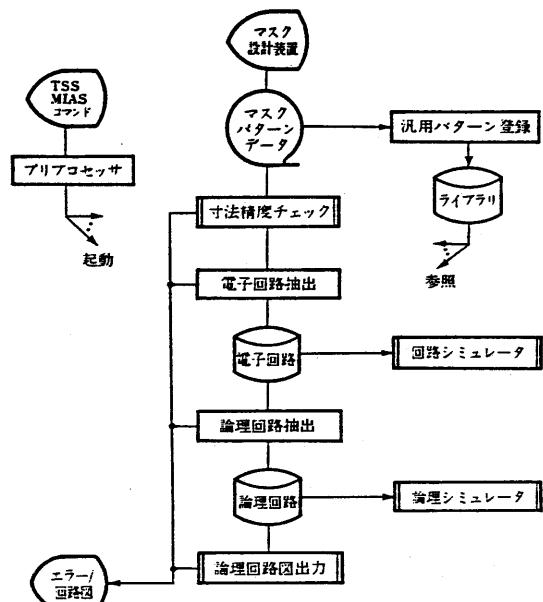


図2 MIAS システム構成図

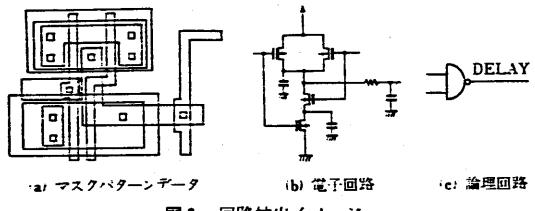


図3 回路抽出イメージ

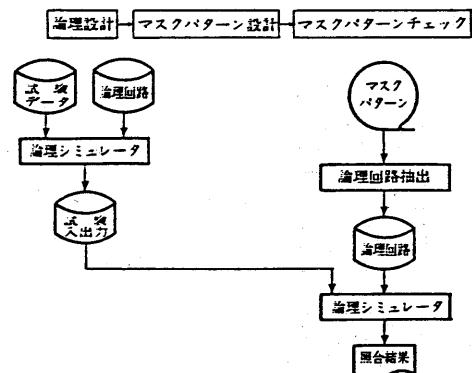


図4 シミュレータによる論理設計誤りの検出

#### 3.5 論理シミュレータ<sup>2)</sup>

抽出された論理回路は、論理シミュレータにより、論理レベルの動作確認を行う。論理設計段階で論理シミュレータによる論理検証を済ませている場合は、その試験データでシミュレートし、結果を比較すれば、機械的に論理誤りを検出できる。

### 3.6 論理回路図出力

抽出された論理回路は、設計者が検証できるように、図面に output する。回路の図面ページへの分割、頁内素子の位置決めおよび素子間経路をすべて自動決定し、A-3図面にプロット出力する（図5）。

### 3.7 汎用パターン登録

すでに正当性が保証されたマスクパターンを組み合せてマスクパターン設計する場合は、その汎用パターンをライブラリに登録することにより、その外部のマスクパターンのみチェックする。階層設計されたマスクパターンでは、下位階層から順次チェックし、ライブラリ登録を繰り返すことにより、VLSI でも短い計算機時間でチェックできる。

### 3.8 プリプロセッサ

ICプリプロセスの相違、チェック項目の相違等により、実行プログラム順序や

必要なファイルがジョブごとに異なる。そのための汎用ジョブ制御言語使用の煩わしさを防ぐため、MIAS コマンドと呼ぶ専用言語を用いる。

- i) プロセスに対応したチェック手順の選択
- ii) チェック項目の選択
- iii) 入出力データの選択

等を入力する。MIAS コマンドは、TSS ターミナルから問い合わせ形式で入出され、TSS にチェックのうえ、汎用ジョブ制御言語に変換される。

### 3.9 エラー出力

シミュレータ以外で検出されたエラーは、マスクパターンと対応した位置にグラフィック出力される（プロッタモードはグラフィックディスプレイ）。そのため、エラー图形とマスクパターンを重ね合わせることにより、設計エラー箇所を、TSS に発見できる。

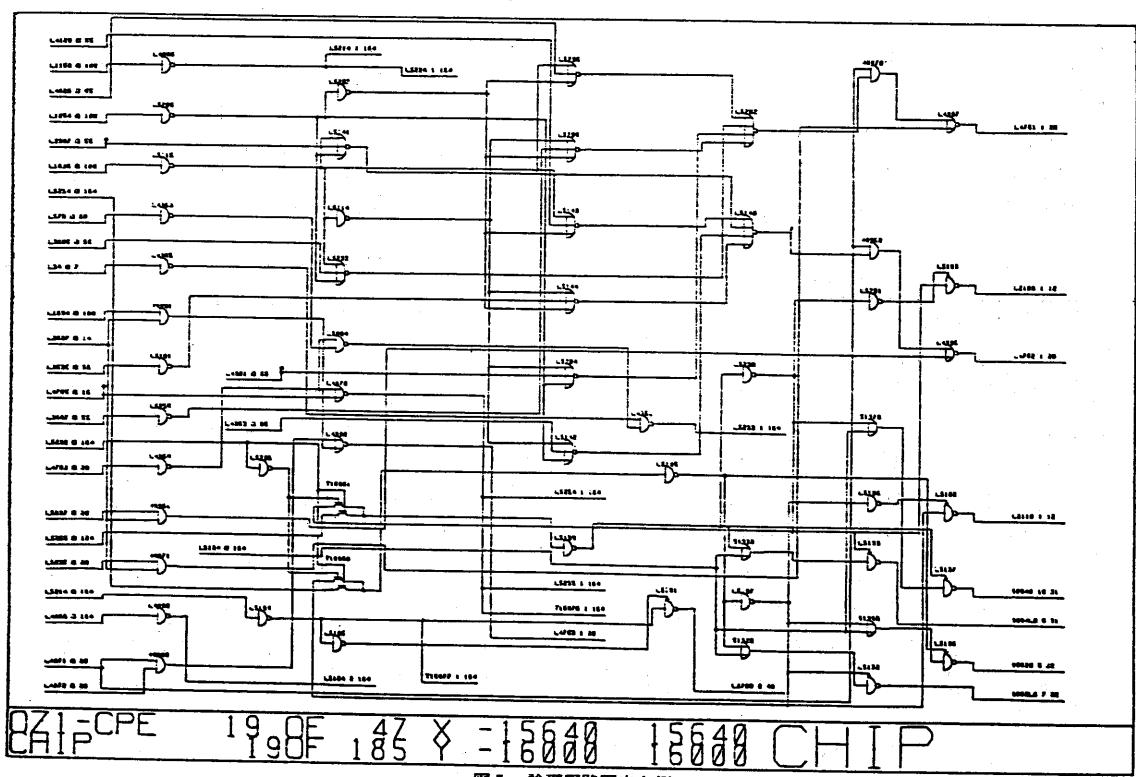


図5 論理回路図出力例

#### 4. 電子回路抽出手法

マスクパターンの層間の重なり関係を調べ、電子回路へ変換する。マスクパターンデータは、層名が付与された閉图形の集合であり、大量图形データを高速に処理するため、マスクパターンデータを層と座標軸成分で分離された枝で記憶し、処理する（図6）。ここで、“枝”とは、图形線分をトップとボトムの各xy座標、左右端、角度等で表現したものである。“サブネット”とは同一層内配線图形である。また、“ネット”とは、スルーハールを介して、層間に配線された图形群であり、ネット番号で識別される。

##### 4.1 層編集

枝データに対して图形演算（図7）を施し、プロセスに対応して手順でサブネット認識、素子のタイプ化を行つ。

〈例1〉セルファラインMOSの拡散サブネットは、拡散層とポリシリコン層をSUB演算することにより求める（図8）。

〈例2〉マスクサイズと仕上ガリサイズの違いは、オーバーサイズ／アンダーサイズを用い仕上ガリサイズに変換する。

##### 4.2 MOS素子認識

Z層間の图形演算を行い、 $\text{「T}_r\text{」}$ や $\text{「C」}$ を認識する。

图形演算結果の枝に以下の枝種別を与える。

- G枝 : 第1入力層枝
- SD枝 : 第2入力層枝
- I枝 : Z層間の枝が内接して  
いるもの
- E枝 : Z層間の枝が外接して  
いるもの

1图形を構成する枝を集め、1图形1素子とみます。

1图形を構成する枝種別から、 $\text{「T}_r\text{」}$ 、 $\text{「C」}$ 、 $\text{「エラー素子」}$ に分類する。

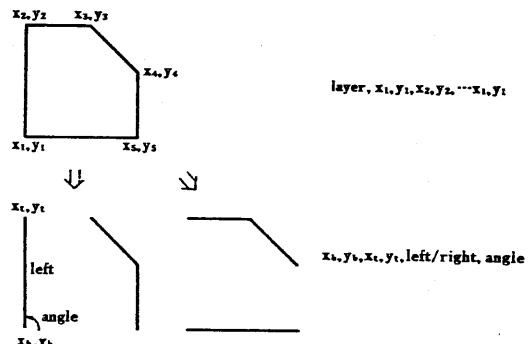


図6 マスクパターンデータの枝データへの変換

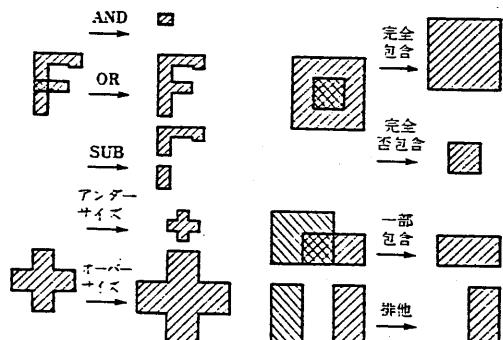


図7 図形演算

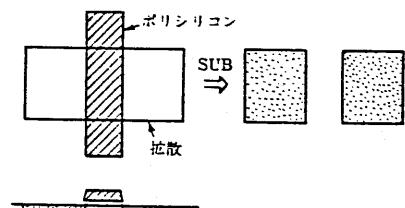


図8 SUB演算による拡散サブネットの認識

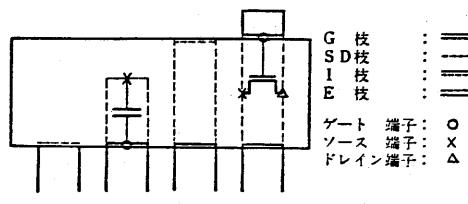


図9 素子形状チェック

- $\text{「T}_r\text{」}$  : G枝群2個、SD枝群2個
- $\text{「C」}$  : G枝群1個、SD枝群1個  
ノーマル素子の場合、G枝上にゲート端子、SD枝上にソース、ドレイン端子を発生する（図9）。

### 4.3 系列解析

認識したサブネット、スルーホールおよび発生した回路素子端子の包含関係を調べ、関係性にする（図10）。

スルーホールを介して接続されたサブネットを集め、ネット番号を与える。  
〈処理手順〉

- ① ネット番号の初期値設定。
- ② ネット番号のカウントアップ。
- ③ 未処理サブネットを1つスタッツに入れ、そのサブネットにネット番号を与える。未処理サブネットがなくなければ④へ。
- ④ スタックから1つサブネットを取り出す。スタックが空なら②へ。
- ⑤ 着目サブネットにスルーホールを介して結ばれるサブネットを全て見つけ、スタッツに入れる。どのときサブネットには、ネット番号を与える。③へ。
- ⑥ 終了。

### 4.4 回路定数算出

素子图形のゲート長、ゲート幅、ゲート面積、配線图形の周囲長、面積から寄生容量、寄生抵抗値を算出する。配線間容量を例に説明する（図11）。

- ① 配線图形を表現する枝にはネット番号が付与されている。
- ② ネット番号の伝搬を考慮しながら图形演算(AND)を行う。
- ③ 演算結果から、1图形を構成する枝を集め面積、周囲長を求める。

### 4.5 パターンデータ処理

#### アルゴリズム

LSIマスクパターンのデータ量は莫大であるため、本アルゴリズムの良否が全体の処理時間を決定する。パターン幅、間隔チェック、图形演算処理、1图形を構成する枝を集める処理(グレーフ化)も、パターンデータが閉图形で構成されるという性質を利用すると、基本的には枝データから图形領域や图形の重なり関係を調べることであ

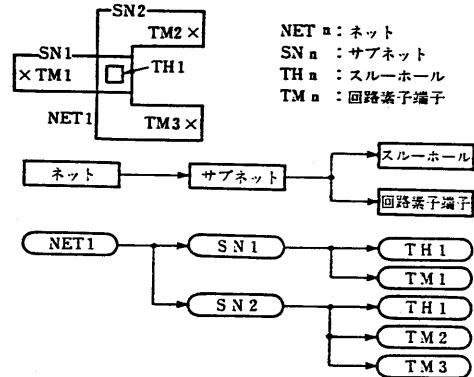


図10 系列解析のイメージ

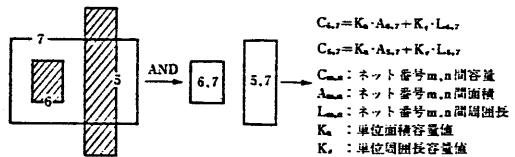


図11 配線間容量算出法

1. 同一のアルゴリズムが使用できる。  
〈処理手順〉

- ① チャンクデータを、ボトムX、Y座標をキーとしてソートする。
- ② 着目するボトムY座標を持つ枝をテーブルに追加入力する。この場合、テーブル内処理済み枝(枝のトップY座標が、着目するボトムY座標未満になった枝)はテーブルから除去しておく。
- ③ テーブル内の枝をボトムX座標をキーとしてソートする。
- ④ カウント $\Delta I$ と、左端処理時に1カウントアップ、右端処理時に1カウントダウントすると決め、ソート順序に従い処理すると
  - a)  $\Delta I = 0$ : 図形領域外
  - b)  $\Delta I = 1$ : 1図形領域内
  - c)  $\Delta I \geq 2$ : 複数図形領域内
 という性質を持つ。この性質を利用して、パターン間隔チェック、图形演算、1图形構成枝のグレーフ化を行う。
- ⑤ 未処理枝が存在すれば②へ、全体処理終了なら⑥へ。

⑥ 終了。

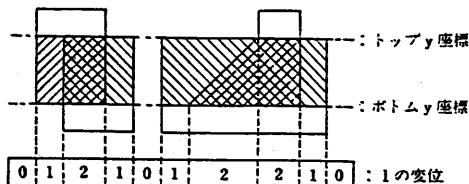


図12 パターンデータ処理アルゴリズム

## 5. 論理回路抽出手法

MOS論理回路は、ゲート信号によるソース、ドレイン間の開閉により論理を構成していこうため、Tr間接続を調べることにより論理ゲートに変換できる。また、遅延時間は出力先の負荷容量の充放電時間であり、寄生容量、抵抗から算出できる。本プログラムは、CMOS, EDMOSを対象としているが、ここでCMOSでの論理ゲート変換およびEDMOS遅延算出手法について説明する。

### 5.1 CMOS論理ゲート変換(図13)

#### 1) Trのグルーピング

任意の未処理Trのドレインを起点として、Trのソース・ドレイン間接続をたどり、アース・電源に至るまでに検出したTrをTrグループとする。

#### 2) 出力信号ネット

##### Trタイプの決定

PタイプTr, NタイプTrのソース・ドレインが混在するネットを信号ネット1(S1)とする。

電源を起点としてTrのソース・ドレイン間接続をたどり、信号ネット1に至るまでに検出したPタイプTrをPタイプドライバTr(DP), 同様にアースを起点として、信号ネット1に至るまでに検出したNタイプTrをNタイプドライバTr(DN)とする。

DP, DNがともに接続する信号ネット1を信号ネット2(出力信号ネット: S2)とする。

信号ネット1の中でDP, DNのいずれか一方レジスタ接続していないネットをリセット付信号ネット(SR)とする。

ソース・ドレン間とともに信号ネット間に接続しているTrを双方向トランジスタTrとする。また信号ネットを起点とするパス上の未認識Trを単方向トランジスタTr(TP, TN)とする。図14に認識例を示す。

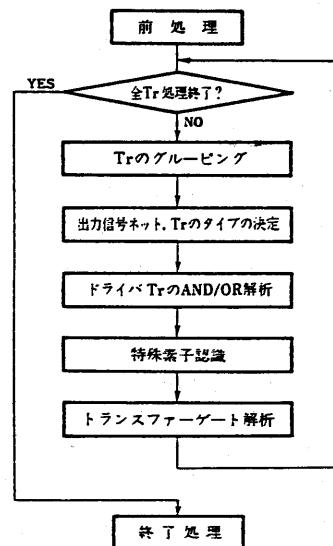


図13 CMOS論理ゲート変換フロー

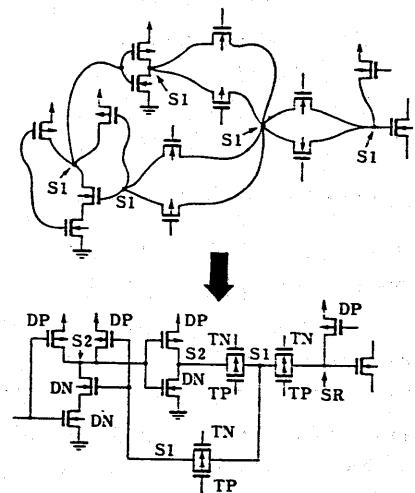


図14 ネットとTrタイプの決定

### 3) ドライバTrのAND/OR 解析

P側、N側をそれぞれ独立に、次の規則に従って、AND/OR 変換する。

P側：シリーズ接続DP →  
OR回路 + DP  
パラレル接続DP →  
AND回路 + DP

N側：ANDとORがP側と逆。

ある出力信号ネットに接続するDP、DNがそれそれを1個に下りまで上の変換を繰り返した後、P側、N側の論理回路を比較する。

一致した場合、N側から変換した回路の最終段を反転する。一致しない場合は、3-State要素を生成する。

インバータの場合にはAND、OR回路は生成されないので1入力NAND要素を生成する。図15に変換例を示す。

### 4) トランジスタゲートの解析

出力信号ネットを起点として、單方向トランジスタTrの接続をたどり、入力側ネットと出力側ネットを決定する。

### 5.2 遅延時間算出

図16(a)に示す電子回路では図16(b)に示す論理回路に変換されるが、この場合の複合ゲート全体の遅延時間は図16(c)のようなモデルで考えることができる。立ち上がり遅延は、スイッチをあわせた場合のC<sub>0</sub>への充電であり、立ち下がりは、スイッチを下した場合のC<sub>0</sub>の放電である。故に、遅延換算のためのパラメータを求め、ユーザーが遅延モデルに従って定義して遅延換算式にパラメータを代入することにより、立ち上がり/立ち下がり遅延を求める。

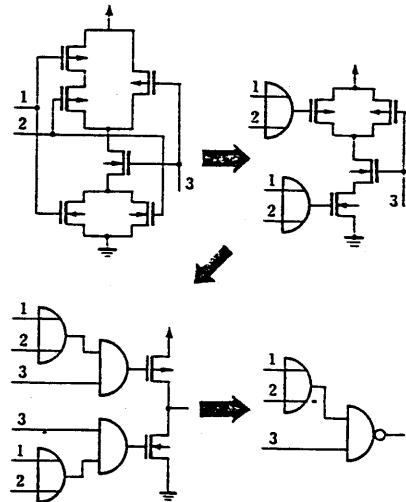


図15 AND/OR 解析

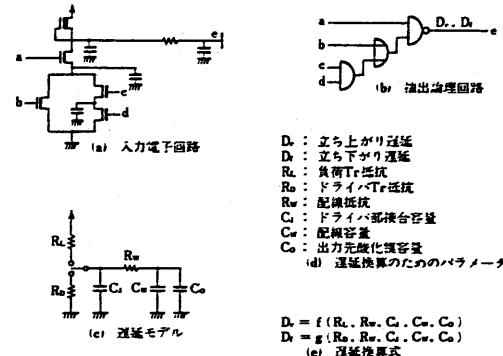


図16 遅延時間計算

### 6. 終りに

本システムは、現在までに約200品種に適用した。主に、

- 論理LSIチップ全体のチェック
- 自動設計のための最下位ブロックの検証
- メモリ周辺回路の動作性能確認等に使用されている。

処理時間は、経験的に、マスクパターンの複雑さにほぼ比例し、10Kゲート論理LSIチップの全項目チェックを実行した場合、約80分(寸法精度チェック、シミュレーションを除く)である(約3 MIPS)。さらに大規模集積回路にも対処可能である。

表 2 評価方法比較

	シミュレータによる評価	試作による評価
プローブ	○任意信号に直のセット、観測ができるため評価が容易。	×内部信号にプローブ端子を設けられない。設けることができても条件が変る。
改良評価	○誤りや改良法は、ただちに修正して評価できる。	×再試作が必要である。
異常入力	○異常入力の条件も評価できる	×異常入力により、回路が壊れことがある。
詳細度	○モデルの作成法で詳細さを任意に変えられる。	×詳細さは、測定器の精度で抑えられる。
	×モデルと実物との差異が問題になる。	○問題なし。
テスト量	×実動作時間より遅いので大量のテストができない。	○大量のテストが可能

MIAS の開発により、数 10 K ゲート級論理回路を論理シミュレータを用い妥当な処理時間と性能を含めてチェックを可能にして。シミュレータによる評価には表 2 に示すようなメリットがある。しかし、抽出回路が計算機を作成してあるため、抽出回路とバーン設計前の回路の対応が付くにくいという問題を残しており、今後改善していく予定である。

## 7. 参考文献

### 1) Alexander:

A Technology Indipendent  
Design Rule Checker,  
USA - JAPAN Computer Con-  
ference Vol. 3, p.p. 412~416  
1978

### 2) 白木 他:

BINALY : 階層構造モデルを  
扱う論理シミュレータ,  
電子装置設計技術研究会資料  
1-3, 1979