

ゲートアレイ設計自動化システム

Gate Array Design Automation System

ゲートアレイでは顧客仕様のLSIを短時間で開発することが要求されている。特に設計工程では、設計品質の向上を図り、かつ多品種のLSIを短時間で設計するため、コンピュータによる設計の自動化が不可欠となっている。日立製作所では、RAM内蔵形CMOSゲートアレイ及びバイポーラとCMOSトランジスタとが同一チップ上に複合化されたBi-CMOSゲートアレイに適用できる設計自動化システムを開発・実用化した。本システムは論理設計から製造・検査用の各種データ作成まで設計工程を幅広くカバーし、充実した論理検証機能、実装設計の自動化、検査用データの評価機能などにより高品質ゲートアレイの短時間開発を実現している。

国友佳男* *Yoshio Kunitomo*
浅野弘道** *Hiromichi Asano*
吉原進** *Susumu Yoshihara*
岩崎孝四郎*** *Kôshirô Iwasaki*
枝川貢**** *Mitsugu Edagawa*
安斉裕臣***** *Hiroomi Anzai*

1 緒言

ゲートアレイは、あらかじめ数百個から数千個のゲートを実装できるベースチップを作っておき、これにユーザーの指定した論理に従ってゲートを実装するセミカスタムLSIであり、各方面に応用されその需要は急増している。最近のゲートアレイへのニーズは、高速、高集積化とともに、RAM(Random Access Memory)の搭載、入出力機能の強化など高機能化へのニーズも高まっている¹⁾。

ゲートアレイでは、多品種のセミカスタムLSIを効率良く開発するために、設計自動化システムが不可欠である。ここに紹介するゲートアレイ設計自動化システムは、RAM内蔵形CMOS(Complementary Metal Oxide Semiconductor)ゲートアレイ、Bi-CMOSゲートアレイ、入出力機能が強化されたゲートアレイなどに汎用的に利用できるシステムである。以下、本システムの構成、機能概要及び特長について述べる。

2 システムの概要

ゲートアレイ設計自動化システムに要求される主な仕様は、(1)短時間でゲートアレイを設計するために、論理情報から自動的にLSIの製造用データと検査用データが作成できること、(2)論理情報が簡単に記述できるなど入力データの作成が容易なこと、(3)設計品質を向上するために、論理情報、製造・検査用データの検証機能が充実していること、(4)目的に合った機能、性能のゲートアレイを設計するために、各種ゲートアレイに適用できること、などである。

日立ゲートアレイ設計自動化システムは、これらの要求仕様を考慮して設計されたシステムである。システムの全体構成を図1に示す。論理設計、実装設計及び診断の三つのサブシステムから成る一貫した設計自動化システムである。論理情報は論理図とテストパターン(論理回路へ与える入力信号列)の情報として入力され、統一した設計データベースをもとに、自動的に論理検証用データ、LSI製造・検査用のデータを作成する。ゲートアレイのLSIに関する各種データは、ライブラリとして統合、管理されている²⁾。

論理設計サブシステムは論理設計をサポートするシステムであり、論理検証用の機械書き論理図、論理シミュレーション結果などを出力する。

実装設計サブシステムは、論理検証が終了した論理図の情報をもとに、LSIチップ上への論理素子の配置と素子間の配線経路を決定し、製造用データを作成する。

診断サブシステムは、検査用データ(テストパターン)の質を向上するために、テストパターンの故障検出率、未検出故障箇所を算出する。また、LSIを検査するためのデータを作成する。

3 論理設計サブシステム

論理設計サブシステムは、(1)統一した記述言語での論理情報の作成、(2)検証容易な機械書き論理図の作成、(3)検証容易

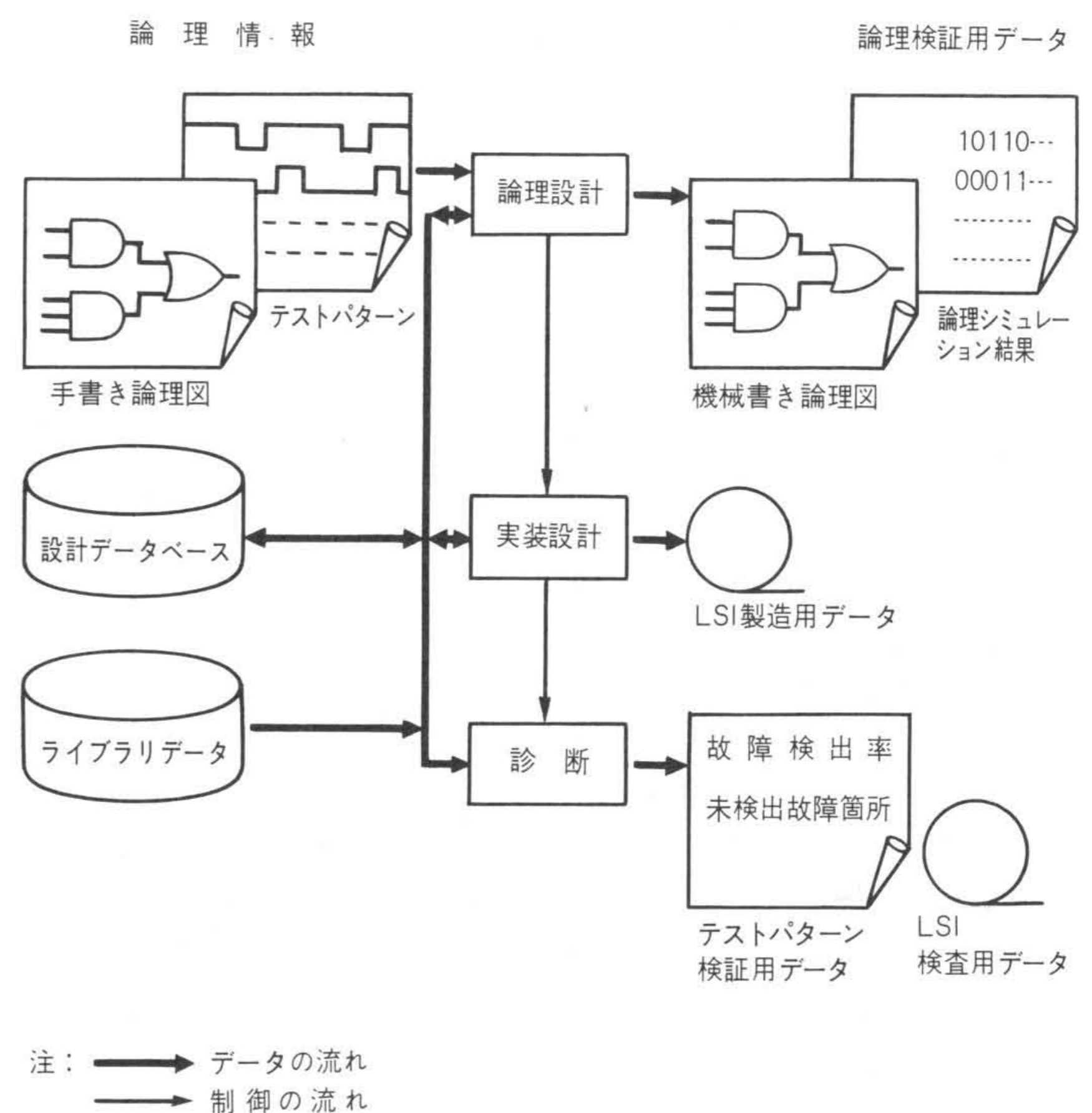


図1 ゲートアレイ設計自動化システムの構成 論理情報を入力し、自動的に論理検証用データ、LSI製造・検査用データを作成する。

* 日立製作所日立研究所 ** 日立製作所武蔵工場 *** 日立製作所高崎工場 **** 日立製作所神奈川工場
***** 日立マイクロコンピュータエンジニアリング株式会社

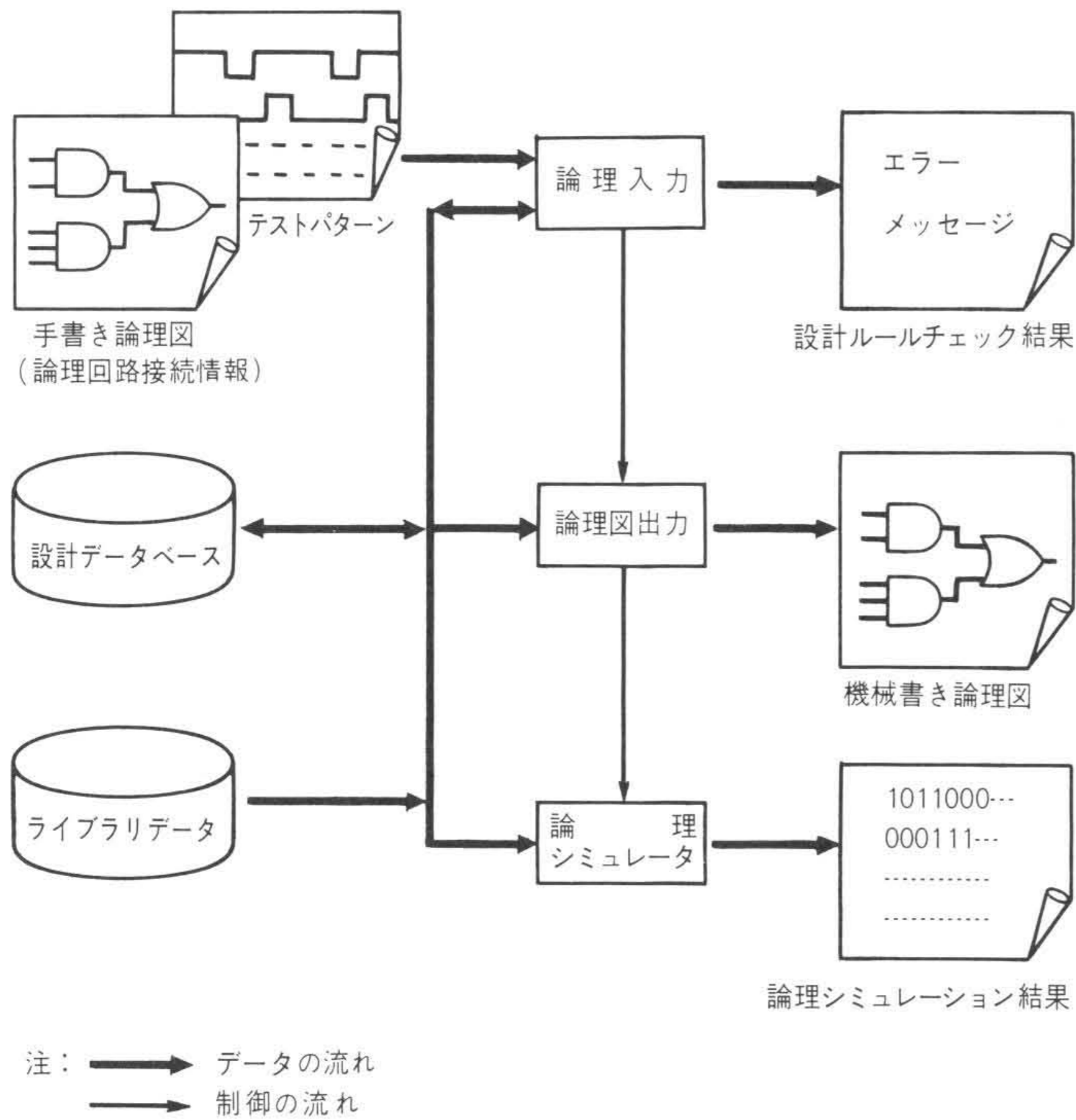


図2 論理設計サブシステムの構成 論理情報を入力し、設計ルールチェック、機械書き論理図の作成、論理シミュレーションを行なう。

な論理シミュレータ機能などにより、設計品質の向上と設計工数の低減をねらったシステムである。

3.1 論理設計サブシステムの構成

本サブシステムの構成を図2に示す。本サブシステムは、論理入力プログラム、論理図出力プログラム、論理シミュレータプログラムの三つから成る。論理入力プログラムは、論理素子間の接続関係を記述した論理回路接続情報(論理図の情報)及びLSI検査用のテストパターンを入力する。入力した情報は、接続チェックやFAN IN/FAN OUT数などのルールチェックによって誤りをなくした後、論理図出力プログラム及び論理シミュレータに入力できるように編集される。論理図出力プログラムは、入力された論理回路接続情報をもとに、論理図を自動作成する。論理検証を行なうための論理シミュレータは、高速処理のためにイベントドリブン方式を採用している³⁾。

3.2 論理回路接続情報の記述言語

論理設計、実装設計、診断の各サブシステムで使われる論理回路接続情報は、共通記述形式であることが望ましい。しかし、必要となる情報には相違があり、すべてを包含するのは記述が複雑となり、デメリットも多い。そこで、共通的部分だけを取り出し、これを言語化した。サブシステム間で共通化できない個別の情報は、各サブシステムで使用するライブラリ情報にもたせて管理する方式をとった。記述言語は、論理素子名(各素子に付けた名称で、通常これが信号名にもなる。)、論理機能名(素子の機能を表わす名称)及び素子に入力している他の素子の論理素子名から構成される。図3に論理素子間の論理回路接続の記述を示す。また、繰り返し現われる論理記述は重複して定義しないで済むように、論理ブロック定義機能(ユーザー定義マクロ)を用意し、記述を簡略化している。論理ブロック定義例を図4に示す。“BEND”の行に記述された“@+Q”と“@-Q”が論理ブロックの出力信号名を表わす。@には論理ブロックを使うときの名称が代入さ

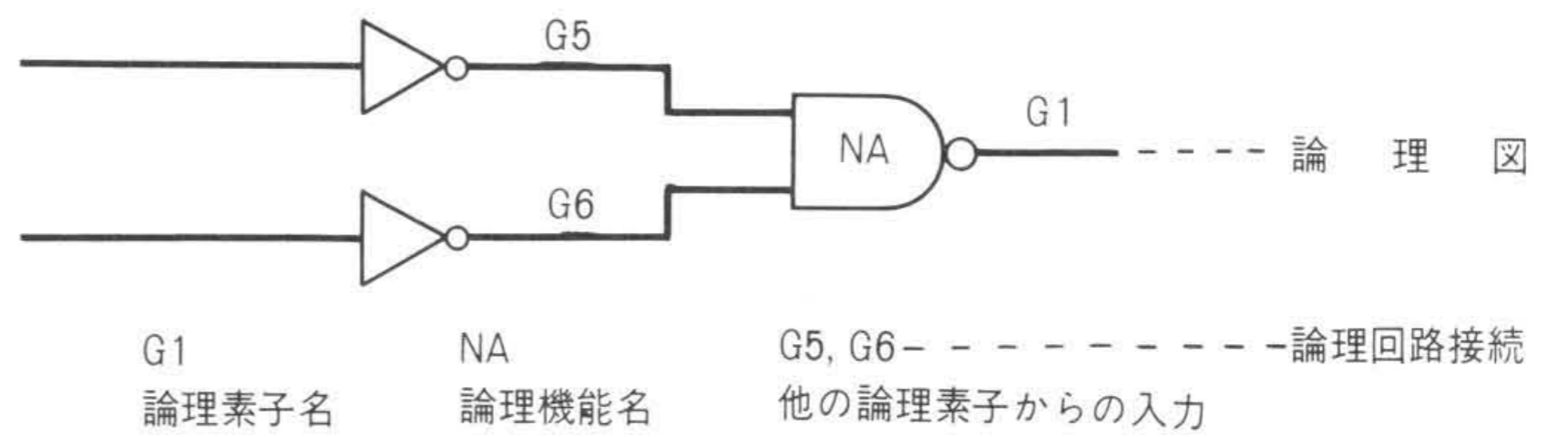
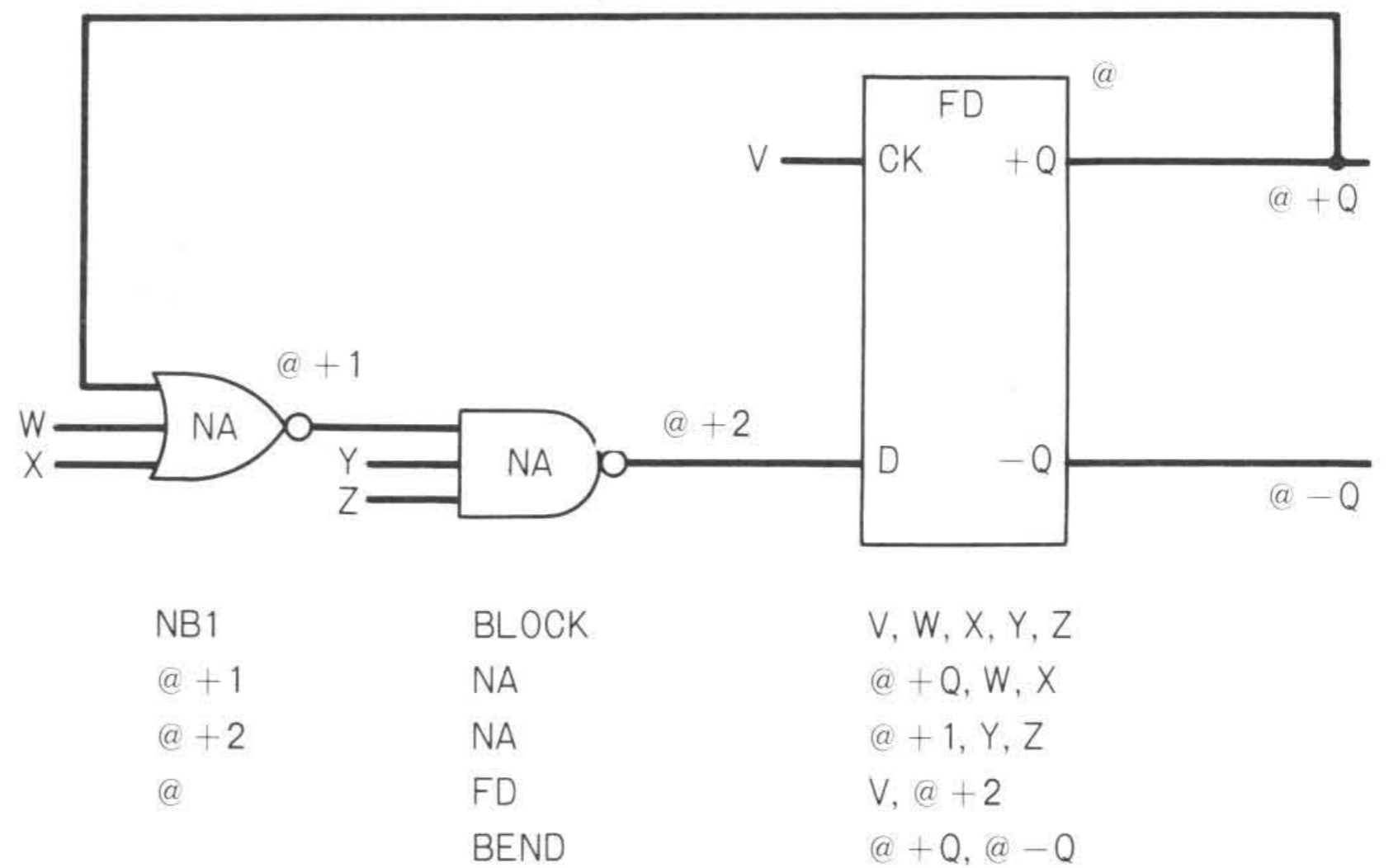


図3 論理素子間の論理回路接続例 G1は論理機能名がNA(論理機能、素子の遅延時間などがライブラリに登録されている。)の論理素子で、入力端子はG5及びG6と接続されている。



注： NB1〔論理ブロック(ユーザー定義マクロ)の名称〕
 V, W, X, Y, Z(論理ブロックの入力信号名)
 @+Q, @-Q(論理ブロックの出力信号名)

図4 論理ブロック定義の例 論理ブロックを構成する論理素子間の接続を記述する。論理ブロック使用時には、入力信号名、出力信号名などは使用時の名前に置き換えられる。

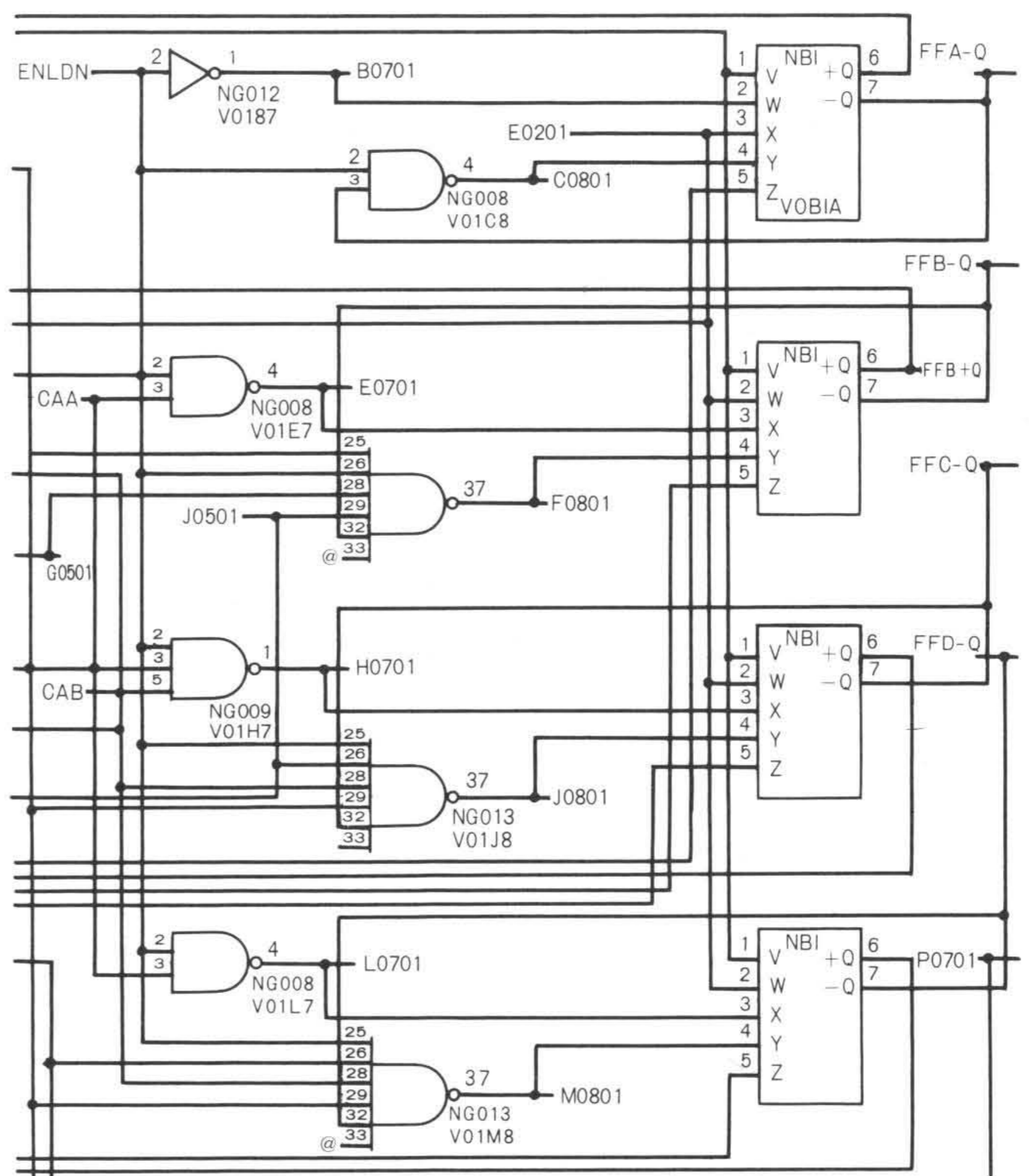


図5 機械書き論理図の例 論理ブロックNB1を含む論理図の一部である。論理素子の位置と接続関係の情報から自動作成される。

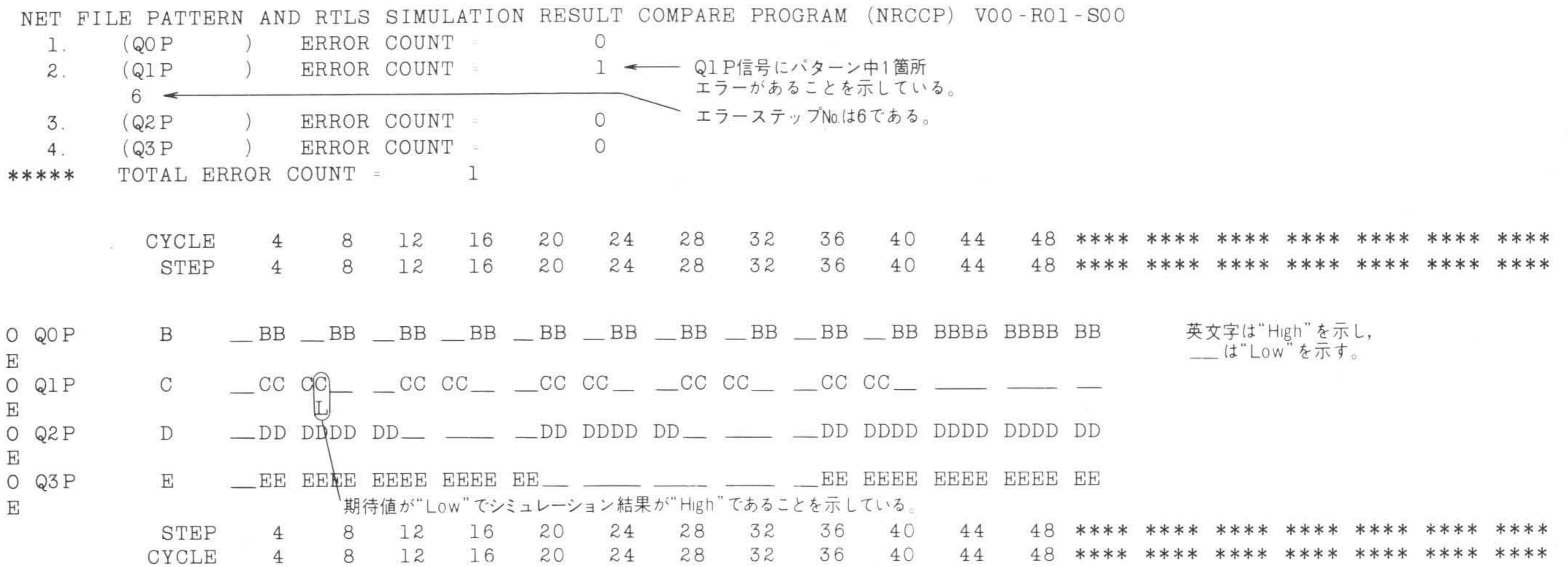


図6 比較チェック結果の例 期待値とシミュレーション結果の不一致の数及び不一致の箇所が分かる。

れる。例えば、論理ブロックを“FFB”という名称で使うと、この名称が“@”に代入され“FFB+Q”と“FFB-Q”が出力信号名になる³⁾。

3.3 機械書き論理図の作成

論理図出力プログラムは、論理素子の位置と接続関係が与えられた論理図の自動作成を行なう。ここでは、見やすさを考慮した配線ルート自動決定、信号名の作画位置決定を行なう。配線手法は線分探索法を使っており、(1)信号単位の概略経路の決定、(2)最適経路の決定の手順で配線が行なわれる。信号単位の概略経路の決定では、n点の端子から同時に複数線分を発見させ、他の端子から延びた線分群と交わりの少ない線分で、かつ自分の端子からの配線長さが最小になる線分を選択する。最適経路の決定では、配線経路の曲がり数が少なく、かつ配線経路長さが短くなるように修正し、冗長のない配線経路とする。論理図の出力例の一部を図5に示す³⁾。

3.4 論理シミュレータの論理検証機能

論理シミュレータは、ユーザーが設計した論理回路に所定の入力信号列を加えた場合、回路を構成する論理素子の出力信号がそれぞれどのような状態になるかを計算する。論理シミュレータの処理結果はリストに出力できるほか、必要な部分を編集しCRT(Cathode Ray Tube)端末に表示できる。この編集表示機能により、論理シミュレータを使っての論理検証が効率良く行なえる。また、論理シミュレータの出力結果とユーザーがあらかじめ設計データベースに入力しておいた論理回路の出力期待値とを比較し、信号名ごとに間違っているテストステップ番号をリスト出力するとともに、その信号の期待値とシミュレーション結果のうち異なるところだけを並記して出力することもできる。比較チェック結果の例を図6に示す。

3.5 DESCの利用

設計作業のうちユーザーに最も身近で、かつ最も時間を要するのが論理設計の検証である。日立製作所では、ユーザーが直接論理設計サブシステムを使用できるようにDESC(Design Engineering Support Center)を国内各地(東京、大阪、名古屋)及び海外(米国、英国、ホンコン)に開設している。DESCではその場で機械書き論理図、論理シミュレーション結果などを提供できるように、ハードウェア面での準備もできている。DESCを利用することにより、ユーザーは迅速に論理検証を進めることができる。また、DESCと半導体工場

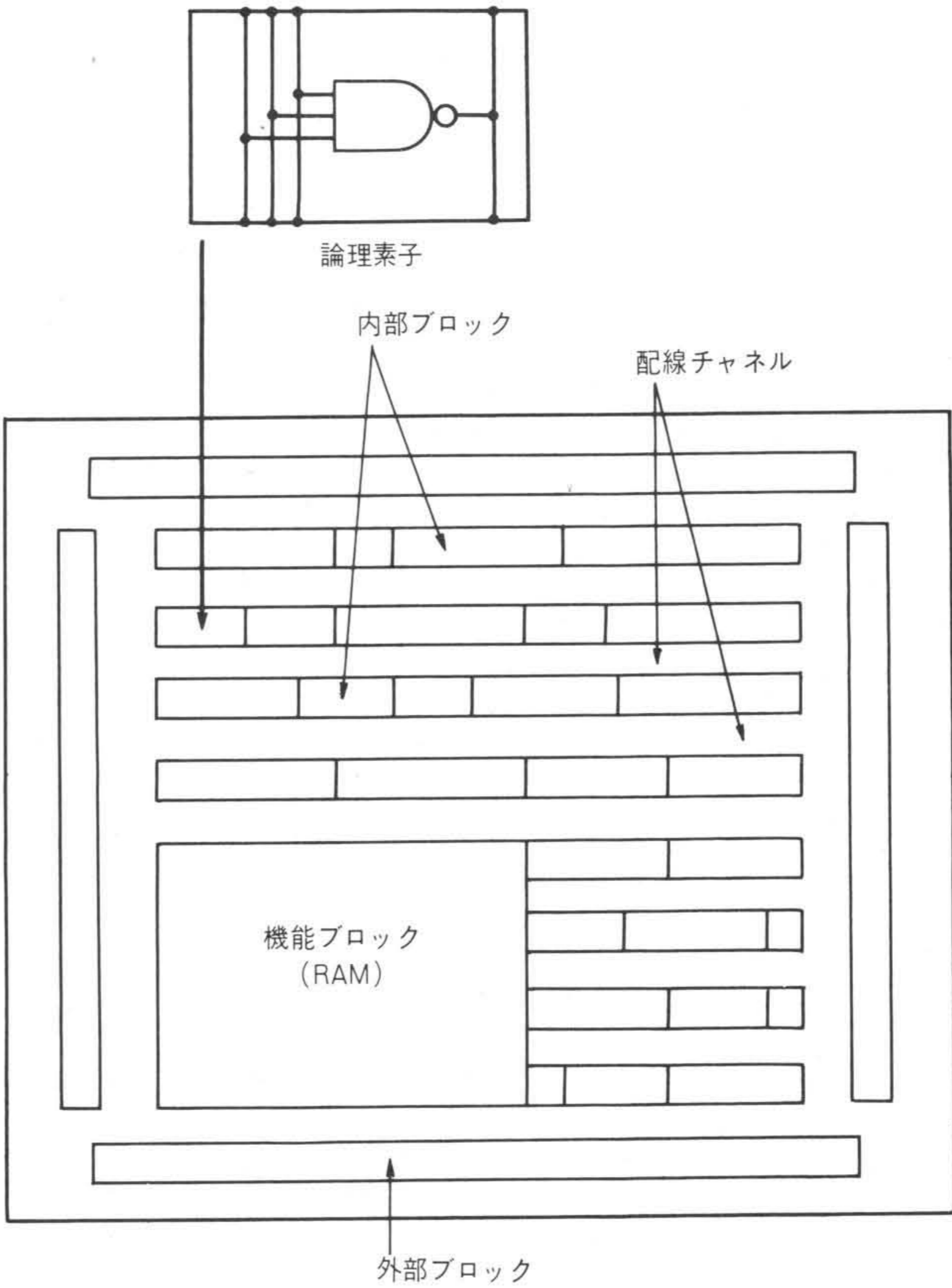
とは通信ネットワークで結ばれていて、論理情報などのデータ伝送ができる。

4 実装設計サブシステム

実装設計では、論理検証が終了した論理回路接続情報をもとにLSIチップ上に論理素子を配置し、素子間の配線経路を決定する。これら一連の作業は大部分が自動化されており、人手による作業を最少限に抑えている⁴⁾。

4.1 自動配置

図7にチップの実装モデルを示す。自動配置では、自動配



注：略語説明 RAM(Random Access Memory)

図7 LSIチップの実装モデル 内部ブロック(論理素子)、外部ブロック(I/Oバッファ)、配線チャンネルから構成される実装モデルを示している。RAMの機能ブロックは、内部ブロックと配線チャンネルを使って実装される。

線での結線率が100%に近づくよう論理素子の配置、交換などが行なわれる。ここでは予想配線長さの最小化、混雑度の平均化が考慮されている。RAMの機能ブロックは、 m ビット× n ワードの指定によりRAMを構成する機能論理素子(アドレスレコーダ、記憶素子など)に自動展開され、配置される。RAM以外の論理情報については、残りの内部ブロック領域を使って素子の配置が行なわれる。したがって、任意サイズのRAMが取り扱えるとともに、RAMの大きさが変わってもチップ全体を有効利用できるようになっている。

4.2 自動配線

自動配線は、自動配置の結果をもとに論理回路接続情報に従って素子間の結線を行なう。RAM内の接続信号は論理信号と併合し、一括配線を行なっている。まず、チップ全体をM×Nの格子に分割し、信号単位に配線順序を決める。次に、この格子を利用して全信号の概略配線経路を決定し、次いで配線チャンネルでの詳細経路を決定する。ここまでの処理で未結線が発生した場合には、更にチップ全体について配線経路を探索し、再配線を行なう。これによりほぼ100%の結線率が得られる。図8は、RAMを含む自動配線結果のX-Yプロット出力図である。もし自動配線で未結線が残った場合には、グラフィックシステムを利用して対話形式で結線修正が行なわれる。この際、配線チェックが、配線の電気的接続関係、実装ルールをチェックし、人手による誤りを未然に防いでいる。以上述べてきたように、実装設計の大部分が自動化されており、設計期間の短縮、信頼性の確保を実現している。

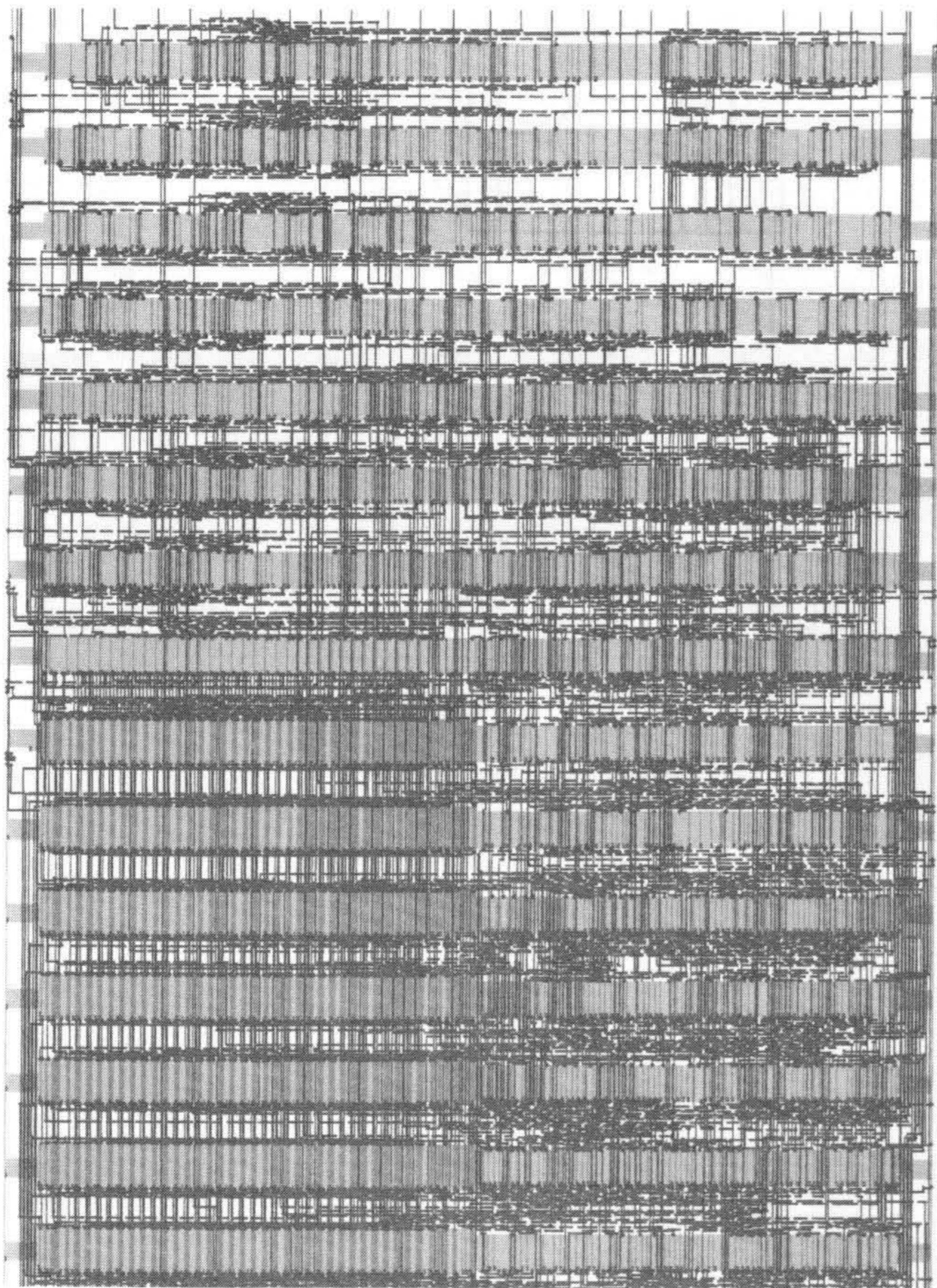


図8 ゲートアレイLSIの配線パターン図 自動配置配線により実装されたゲートアレイLSIのパターン図である。左下は自動生成されたRAM部分である。

// UNDETECTABLE FAULTS_GATE //

LINE	ELEMENT1	ELEMENT2	0	1	LINE	ELEMENT1	ELEMENT2	0	1
41.	J4	_DABLO+I	.	1	44.	J5	_DACLO+I	.	1
65.	ENLDN	_H0701+1	.	1	72.	ENATI	_P0601+I	.	1
81.	CAB	_J0501+I	.	1	82.	CAA	_J0501+I	.	1
91.	UDN	_J0601+I	.	1	92.	ENATI	_Q0601+I	.	1
103.	ENLDL	_J0801+I	.	1	108.	??FH	_J0801+I	.	1
134.	FFA+1	_FFA+2+I	.	1	148.	FFA+EO	_FFA+I	0	.
154.	EO201	_FFB+1+I	.	1	171.	FFB+EO	_FFB+I	0	.
177.	EO201	_FFC+1+I	.	1	194.	FFC+EO	_FFC+I	0	.
200.	EO201	_FFD+1+I	.	1	205.	DADLO	_FFD+2+I	.	1
219.	FFD+1	_FFD+D	0	.					

44

FAULTS DETECTION RATE(GATE) = 74.32 = 110 / 148

図9 故障シミュレータの出力結果の例 未検出故障箇所は論理素子間の接続で示され、0/1は0/1縮退故障が未検出であることを示す。

5 診断サブシステム

ユーザーが論理設計段階で用意したテストパターンでは、論理回路のすべての故障検出ができない場合がある。故障検出率を求めるために、診断サブシステムには故障シミュレータが用意されている。この故障シミュレータは、回路のすべての論理素子及び配線に単一の“0”又は“1”の縮退故障(論理的に等価な故障は除く。)を仮定し、テストパターンにより検出される故障を求め、検出故障数/仮定故障数から故障検出率を算出する。また、仮定した故障のうち検出されない故障を出力する。図9に、故障シミュレータの出力結果の例を示す。未検出故障個所の情報により、その箇所の故障が検出できるようにテストパターンを追加して故障検出率を向上し、LSIの検査の質を向上することができる。

6 結 言

以上述べたように、ゲートアレイ設計の各工程に設計自動化プログラムが用意されており、論理検証、テストパターンの検証により設計品質を向上できる。また、検証が終了した情報から自動的にLSI製造・検査用の情報が作成でき、多品種の開発に対応できるようになっている。本システムは、RAM内蔵形CMOSやBi-CMOSのゲートアレイ設計に適用されている。今後も半導体技術の向上により、ゲートアレイの高機能化や集積度の増大が続くと考えられるので、自動化技術の拡充を促進して開発力のいっそうの強化に努力してゆきたいと考えている。

参考文献

- 1) 山村：市場が急成長し始めたゲート・アレイ，日経エレクトロニクス，111～122(1983.2.28)
- 2) 児島，外：ゲートアレイ用総合CADシステム-システム構成-，情報処理学会第27回全国大会(1983)
- 3) 菅原，外：ゲートアレイ用総合CADシステム-論理設計システム-，情報処理学会第27回全国大会(1983)
- 4) 国友，外：ゲートアレイ用総合CADシステム-レイアウト・システム-，情報処理学会第27回全国大会(1983)