

Title	論理回路のテスト容易化設計に関する研究
Author(s)	三浦, 幸也
Citation	大阪大学, 1992, 博士論文
Version Type	VoR
URL	https://doi.org/10.11501/3087936
rights	
Note	

Osaka University Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

Osaka University

論理回路の
テスト容易化設計に関する研究

1992年1月

三浦 幸也

内容梗概

近年の半導体技術の向上により、集積回路(IC)の大規模化、高機能化が一段と進んでいる。その結果、故障が増加するのみでなく、テストも複雑かつ困難になり、また、ICの製造コストに対するテストのコストの占める割合が高くなってきている。さらに、CMOS技術が半導体ICの主流になるにつれ、これまで考えられてきた縮退故障モデルでは説明できない故障が多く現れるようになってきた。その代表的なものとしてMOSトランジスタの開放故障、導通故障、ゲート酸化膜短絡故障などが考えられている。

メモリのテストはデータの書込み、読出しによって行われる。メモリの記憶容量を N とし、テストに必要なアクセス回数のオーダーは $O(N^2)$ 、 $O(N^{3/2})$ 、 $O(N)$ に大別できる。今後、さらに記憶容量が増加するにつれ、テスト時間の短い $O(N)$ のテスト系列を用いてもテストに時間がかかり過ぎ、その適用は困難になるものと考えられる。現在、メモリに対し、複数のメモリセルを同時にテストできるようにテスト容易化設計を行い、テスト時間を短縮させる方法が提案されている。また、組込みテストを用いるテスト方式もいくつか報告されている。しかし、いずれもメモリの回路変更や単にテスト機構をチップ内に組込んだだけのものである。テスト系列長は $O(N)$ であり、テストに時間がかかるという問題が依然として残っている。

CMOS回路内のトランジスタの開放故障を検出するためには、連続した2つのテストベクトルからなるテストが必要であることが知られている。しかし、入力信号のタイミング・スキューや2つのテストベクトル間でのハザードの発生のため、開放故障の検出が不可能となる場合もある。これに対処するため、ローバストテストが提案されているが、そのテスト生成は困難であり、また、回路構成によっては必ずしもローバストなテストベクトルが得られるとは限らない。さらに、開放故障に対するテスト容易化設計もいくつか提案されている。しかし、いずれも1ゲート当たり2~4個の付加トランジスタが必要であり、また、付加トランジスタの駆動能力を考慮した場合には、かなりの面積オーバーヘッドになるものと考えられる。

正常な完全相補形CMOSゲートから成る回路では、回路動作が安定状態にあるとき電流

はほとんど流れない。これに対し、トランジスタの導通故障が顕在化したならばかなり大きな電流が流れることから、電圧源の静的電流を計測し故障検出を行う電流テスト法が提案されている。一般に外部テスト装置で電流計測を行う場合、テスト装置自身の負荷がICのそれより大きいことや、ICの出力ドライバが比較的大きな電流を流すことなどの理由により、IC本来の動作クロック周期で電流計測を行うことは困難である。また、電流計測回路をIC内に組込んだ組込み電流テスト法が提案されている。しかし、これまで提案されている回路は動作速度は速いが、その動作マージンが小さいためICの製造ばらつきを考慮した場合、故障検出能力が低下するものと考えられる。また、電流テストでは観測点は電圧源端子であるため、導通故障を顕在化するテストベクトルのみを印加するだけで十分である。よって、従来の縮退故障のテストに比べ、テストベクトル数が少なくなることが知られている。このため、テスト時に動作クロック周期を遅くして電流テストを行うことが提案されているが、IC本来の動作クロック周期より遅いクロック周期でテストした場合、真にその回路のテストを実施したことになるのかという疑問が残る。

本論文ではテスト容易化設計手法を取り入れ、これらの問題を解決するテスト方式について検討した。対象とする回路は論理回路であり、メモリの組込みテスト、CMOS組合せ回路のトランジスタの開放故障に対するテスト容易化設計、組込み電流テストについて論じる。

メモリについては、1ワード線上のすべてのセルを同時にテストできる組込みテスト方式を提案する。CMOS組合せ回路については、トランジスタの開放故障に対し、ゲート出力を直接アクセスするローバストテスト方式を提案する。さらに、組込み電流テストのための回路設計とテスト生成について提案する。これは単位時間内の電流の積分値を計測し、その値により正常/故障を判定する回路であり、また、この回路に適した動的電流値がほぼ等しいテスト系列の生成を行う。

第1章では、論理回路のテストの問題点とこれまで行われてきた研究結果について述べる。また、本研究の目的とその方針について述べる。

第2章では本論文の基礎となる論理回路のテスト方法、基本的概念について述べる。テスト方法の分類、テスト容易化設計の概念と設計方式などについて述べる。また、メモリのテスト、組合せ回路のテストについては、これまでに行われてきた代表的なテスト方法について述べる。最後に電流テスト法の特徴について述べる。

第3章ではメモリの組込みテストについて論じる。組込みテスト法では、被テスト回路内にテスト用回路を組込むため、観測・制御点を回路内に任意に設けることができ、外部テスト法では不可能な方法でテスト容易化をはかれるものと考えられる。本論文では、この考えに基づき、メモリの新しい組込みテスト方式を提案する。これは、メモリの内部動作、特に、セルアレイの動作に着目し、1本のワード線上のすべてのセルの同時テストを可能としたテスト方式である。また、この方式に適したテスト手続き、テスト用回路をも提案した。対象故障として、その検出が最も困難であると考えられているパターン依存故障、および、デコーダ、読出し・書込み回路の故障を考える。これらの故障に対し、提案したテスト方式を用いることにより $O(N^2)$ のテスト系列でテストが可能となり、大幅なテスト時間の短縮につながる。パターン依存故障に対し、提案したテスト手続きは $302N^{1/2}$ の準最適なテスト系列を生成する。さらに、本テスト方式に必要なテスト用回路は、シフトレジスタ、カウンタなどを用いて比較的簡単な機構で実現でき、しかもそのハードウェア量はメモリの記憶容量が増加するにつれて無視できるほど小さくなる。

第4章では、CMOS 組合せ論理回路のトランジスタの開放故障に対するテスト容易化設計の提案とテスト生成について論じる。本論文では、開放故障が顕在化されたとき、ゲートは見かけ上トライ・ステート素子になることを利用し、開放故障を顕在化させるテストベクトルを印加したあとに回路外部からゲート出力をアクセスするテスト方式の提案を行う。この方式はハザードなどの影響を受けずに開放故障のテストを行えるローバストテストである。本方式は縮退故障にも適用できる。さらに、回路内部の再収斂ゲート出力点をテスト点とし、テスト点の減少をはかる。これによりテストのローバスト性を失うことなくテスト点数を大幅に減少させることができ、付加ハードウェア量の減少をはかれるものと考えられる。テスト生成の実験を行い、本テスト方式によるローバストテストに必要なテストベクトル数を求める。

第5章では、組込み電流テストのためのテスト用回路の提案とその回路に適したテスト生成について論じる。CMOS 回路において本質的に生じる動的電流の影響を少なくするために、提案したテスト用回路は動的電流を含んだ電流計測、すなわち、単位時間内の電流の積分値を計測し、その値により正常/故障の判定をする。この計測方式により比較的分解能が高く、かつ、高速の電流テストが行える。SPICE3 シミュレータで回路シミュレーションを行い、要求した機能を満足する回路であることを確認する。さらに、テスト回路の構

成は比較的簡単で、かつ、その自己テストも可能である。レイアウト設計を行い、回路を実現するために必要なチップ面積を求める。また、提案した回路に適したテスト系列の生成手法について検討する。これは、テスト系列中のすべてのテストベクトル間での動的電流値をほとんど等しくするために、任意の連続する2つのテストベクトル間でのゲート出力値の変化数がほぼ等しいテスト系列を生成することを意味する。このテスト系列は従来の縮退故障に対するテスト生成プログラムの一部を利用して生成でき、また、従来手法による縮退故障に対するテストベクトル数より少ないテストベクトル数となる。

第6章では、以上の研究結果をまとめ、提案した種々のテスト容易化設計手法の特徴と成果について述べる。また、今後の課題についても言及する。

目次

第1章 序論	1
第2章 論理回路のテストの基本概念	7
2.1 故障モデル	8
2.2 テスト方法	8
2.3 テスト容易化設計	10
2.4 メモリのテスト	13
2.5 組合せ回路のテスト	14
2.6 電流テスト	23
2.7 総括	26
第3章 メモリの組込みテスト	27
3.1 故障モデルと検出条件	28
3.1.1 故障モデル	28
3.1.2 検出条件	32
3.2 テスト方式	33
3.2.1 セルアレイ	33
3.2.2 デコーダ, 読出し・書込み回路	35
3.3 テスト系列	36
3.3.1 セルアレイのテスト系列	36
3.3.2 デコーダ, 読出し・書込み回路のテスト系列	43
3.4 出力系列の圧縮	43
3.4.1 セルアレイのデータ圧縮	43
3.4.2 デコーダ, 読出し・書込み回路のデータ圧縮	47

3.5	テスト用回路	49
3.5.1	セルアレイのテスト用回路	49
3.5.2	デコーダ, 読出し・書込み回路のテスト用回路	50
3.6	評価	50
3.6.1	テスト能力	50
3.6.2	テスト時間	51
3.6.3	付加ハードウェア量	52
3.7	総括	53
第4章	組合せ回路の開放故障のテスト容易化設計	55
4.1	故障モデルと検出条件	56
4.2	開放故障のテスト法	57
4.2.1	ローバストテスト手法	57
4.2.2	実装方法	59
4.2.3	テスト点の選択	63
4.3	テスト生成	66
4.3.1	テスト生成アルゴリズム	66
4.3.2	実験結果	68
4.4	総括	69
第5章	組合せ回路の組込み電流テスト	71
5.1	故障モデルと検出条件	72
5.2	テスト用回路	75
5.2.1	機能設計	75
5.2.2	回路設計	77
5.2.3	テスト用回路の自己テスト	85
5.2.4	レイアウト設計	86
5.3	テスト生成	86
5.3.1	テスト系列と回路動作	86

5.3.2	テスト系列生成手法	90
5.3.3	テスト生成アルゴリズム	95
5.3.4	実験結果	97
5.4	総括	98
第6章	結論	99
謝辞	103
参考文献	105
論文リスト		

第 1 章

序論

近年の半導体技術の向上により，集積回路(IC)の大規模化，高機能化が一段と進んできている．その結果，故障が増加するのみでなく，テストも複雑かつ困難になり，また，ICの製造コストに対するテストのコストの占める割合が高くなってきている．ここで，テストとは，与えられた回路の故障の有無を調べることを意味している．

ICのテストは，通常，外部テスト装置から被テスト回路(Circuit Under Test)の外部入力端子にテストベクトルを印加し，その出力応答を被テスト回路の外部出力端子で観測し故障の有無を判断する．このため，ICのテストの困難さは，テストの制御・観測点がICの外部入出力端子のみに制限されることに由来する．一般に，ICの集積度が増加しても外部入出力端子はそれほど増加しないため，1入出力端子当りの素子数が増大し，テストがより困難になってきている．その結果，大量のテストベクトルを印加しなければICのテストができなくなり，テスト生成に要する時間や，テストすること自体に要する時間が大幅に増加している．

これらの問題を解決するために，従来，テスト生成の高速化・高効率化の研究が行われてきているが，これは，与えられた回路をテストするという立場である．これに対し，回路の設計段階からテストを考慮し，テストの容易な回路を設計するという立場であるテスト容易化設計⁽¹⁾が提案されている．テスト容易化設計は被テスト回路の可制御性・可観測性を向上させる設計手法である．その目標は，テスト生成の高速化，テスト時間の短縮であるが，このトレードオフとしてテスト容易化のための付加ハードウェアの増加，動作

速度の低下などが問題となる。また、テスト容易化設計手法の一つにテスト機能をICチップである被テスト回路内に組込む、組込みテスト法⁽¹⁰⁻¹³⁾がある。これはICチップ自身がテスト機構を備えているため、大がかりなテスト装置の不要、保守の容易性などの利点がある。またこの他に、外部テスト法では不可能なテスト方法でテストできるという特徴もある。

CMOS技術が半導体ICの主流になるにつれ、これまで考えられてきた縮退故障モデルでは説明できない故障が多く現れるようになってきた。故障解析の結果、その代表的なものとしてMOSトランジスタの開放故障 (Stuck-Open Fault), 導通故障 (Stuck-On Fault), ゲート酸化膜短絡故障 (Gate Oxide Short Fault), pn接合不良, 短絡故障 (Bridging Fault)⁽¹⁴⁻²²⁾などが考えられている。これらの故障に対するCMOS ICの新しいテスト手法として電流テスト法^(16-18, 21, 22)が提案されている。これは、CMOS回路では回路が安定状態にあるとき、正常ならば電流がほとんど流れないことに着目したテスト手法である。

本論文はICのテスト容易化設計に関する研究についてまとめたものである。対象とする回路は論理回路であり、メモリの組込みテスト、CMOS組合せ回路のトランジスタの開放故障に対するテスト容易化設計、組込み電流テストのための回路設計について論じる。

メモリについては、1ワード線上のすべてのセルを同時にテストできる組込みテスト方式を提案する。CMOS組合せ回路については、トランジスタの開放故障に対し、ゲート出力を直接アクセスするローバストテスト方式を提案する。さらに、組込み電流テストのための回路設計とテスト生成について提案する。これは単位時間内の電流の積分値を計測し、その値により正常/故障を判定する回路であり、また、この回路に適した動的電流値がほぼ等しいテスト系列の生成を行う。なお、本論文で扱う論理は正論理のみである。

メモリのテストはデータの書込み、読出しによって行われる。メモリの記憶容量を N とし、対象とする故障により、テストに必要なアクセス回数のオーダーは $O(N^2)$, $O(N^{3/2})$, $O(N)$ に大別できる⁽²³⁻³⁴⁾。今後、さらに記憶容量が増加するにつれ、 $O(N^2)$, $O(N^{3/2})$ のテストに比べ故障検出能力は劣るがテスト時間の短い $O(N)$ のテストを用いてもテストに時間がかかり過ぎ、その適用は困難になるものと考えられる。

メモリに対し、複数のメモリセルを同時にテストできるようにテスト容易化設計を行い、テスト時間を短縮させる方法が提案されている^(35,36)。また、組込みテスト法を用いるテスト方式もいくつか報告されている⁽³⁷⁻⁴³⁾。しかし、いずれもメモリ回路の回路変更や単

にテスト機構をチップ内に組込んだだけのものである。テスト系列長は $O(N)$ であり、テストに時間がかかるという問題が依然として残っている。

組込みテスト法では、被テスト回路内にテスト用回路を組込むため、観測・制御点を回路内に任意に設けることができ、被テスト回路の外部からではテスト不可能な方法でもテストを行うことができ、テスト容易化、特に、テスト時間の短縮がはかれるものと考えられる。本論文では、この考えに基づき、メモリの新しい組込みテスト方式を提案する⁽⁴⁴⁾。これは、メモリの内部動作、特に、セルアレイの動作に着目し、1本のワード線上のすべてのセルの同時テストを可能としたテスト方式である。この方式により $O(N^{1/2})$ のテスト系列でテストが可能となり、大幅なテスト時間の短縮につながる。また、メモリの主要な構成回路であるセルアレイとデコーダ、読出し・書込み回路とのテストを分離して行うため、テスト能力が一層高まるものと考えられる。さらに、組込まれるテスト用回路は、比較的容易に構成でき、しかもそのハードウェア量はメモリの記憶容量が増加するにつれて無視できるほど小さくなる。

次に、CMOS 組合せ論理回路のトランジスタの開放故障を対象としたテスト容易化設計について述べる。開放故障を検出するためには、連続した2つのテストベクトルからなるテストが必要であることが知られている⁽⁴⁶⁻⁵⁰⁾。しかし、入力信号のタイミング・スキュー (Timing Skew) や2つのテストベクトル間でのハザード (Hazard) の発生のため、開放故障の検出が不可能となる場合もある^(49,51,52)。これに対処するため、ローバストテスト (Robust Test) が提案されているが^(49,51,52)、そのテスト生成は困難であり、また、回路によっては必ずしもすべての故障に対してローバストテストを実現できるとは限らない。

一方、開放故障に対するテスト容易化設計もいくつか提案されている^(51,53-56)。しかし、いずれも1ゲート当たり2～4個の付加トランジスタが必要であり、また、付加トランジスタの駆動能力を考慮した場合、かなりの面積オーバーヘッドになるものと考えられる。

これまでに提案されている開放故障に対するテストのほとんどは、故障が顕在化されたときの負荷容量内の充放電電圧を利用しているため、その周囲の影響を受け易いという欠点がある。本論文では、故障を顕在化させるテストベクトルを印加したあとにゲート出力をアクセスし故障の有無をテストする、ローバスト性を考慮したテスト方式を提案する⁽⁵⁷⁾。また、内部テスト点を減らすための一つの選択方法を示す。一般には、すべてのゲート出力に対し本テスト方式を適用する必要がなく、再収斂ゲートの出力をテスト点とすれ

ばローバストテストが可能であることがわかった。これにより付加ハードウェア量を減らすことができる。さらに、クロスチェック・テスト技術 (CrossCheck Test Technology)⁽⁵⁹⁾ で用いられているクロスポイント・スイッチ (Cross-Point Switch) を導入することで、付加ハードウェア量を一層減少させることができる。付加面積オーバーヘッドを算出するために、加算器に対してレイアウト設計を行う。また、本方式を用いてローバストテストを実現するために必要なテストベクトル数を求めるために、テスト生成の実験を行う。

さらに、組込み電流テスト法による CMOS IC の故障検出について述べる。MOS トランジスタの固有の故障である開放故障、導通故障、ゲート酸化膜短絡故障などは、その存在時にも回路は論理的に正常な動作をする場合があるため、従来の縮退故障を対象としたテストでは、これらの故障の検出を完全に保証できない。

正常な完全相補形 CMOS ゲートから成る回路では、回路動作が安定状態にあるとき、その pMOS トランジスタ回路と nMOS トランジスタ回路の構成からゲート出力は常に V_{DD} 、GND のいずれか一方にのみ接続されている。よって、回路動作が安定状態にあるとき電流 (以下、静的電流) がほとんど流れないのに対し、縮退故障を含めて前述した故障が顕在化したならば、かなり大きな静的電流が流れることが多い。従って、電圧源の静的電流を計測することにより CMOS 回路のテストを行うことができる^(16-18,21,22)。これは電流テスト、または、 I_{DDQ} テストと呼ばれている。一般に外部テスト装置で電流計測を行う場合、テスト装置自身の負荷が IC のそれより大きいことや、IC の出力ドライバが比較的大きい電流を流すことなどの理由により、IC 本来の動作クロック周期で電流計測を行うことは困難である⁽⁶⁰⁻⁶²⁾。この問題は CMOS IC ではゲートの出力論理の切替わり時、すなわちスイッチング時に電流 (以下、動的電流) が流れるため、IC 本来の動作クロック周期で、これと静的電流との弁別が困難であることに起因している。一方、電流計測回路、いわゆる電流計を IC 内に組込んだ組込み電流テスト法⁽⁶⁰⁻⁶²⁾ が提案されている。しかし、これまでに提案されている回路^(60,61) は動作速度は速いが、その動作マージンが小さいため IC の製造ばらつきを考慮した場合、故障検出能力が低下するものと考えられる。

電流テストでは観測点は電圧源端子 V_{DD} または GND であるため、故障を顕在化するテストベクトルのみを印加するだけで十分である。すなわち、従来の縮退故障を対象とした経路活性化法によるテストでは、故障の顕在化と故障の影響の外部出力までの伝搬とが必要であったのに対し、電流テストでは故障の顕在化のみで十分である。よって、縮退故障

のテストに比べ、テストベクトル数が少なくなることが知られている^(62,63-65)。このため、速い動作クロック周期に対して、電流テストの適用は困難であるという問題に対処するために、テスト時に動作クロック周期を低くして電流テストを行うことが提案されているが^(18,62,63,65)、IC本来の動作クロック周期より遅いクロック周期でテストした場合、真にその回路のテストを実施したことになるのかという疑問が残る。

本論文では、動的電流を含んだ電流計測、すなわち、単位時間内の電流の積分値を計測する回路を提案する^(66,67)。提案する回路に対してSPICE3シミュレータ^(68,69)で回路シミュレーションを行い、その機能を確認する。また、レイアウト設計を行い、提案する回路を実現するために必要なチップ面積を求める。さらに、提案する回路に適したテスト系列の生成手法について述べる。これは、動的電流のばらつきを少なくするために、2つのテストベクトル間でのゲート出力値の変化数がほぼ等しくなるようにしたテスト系列を生成する。テスト系列は従来の縮退故障のテストのアルゴリズムの一部を利用して生成でき、縮退故障のテストよりも少ないテストベクトル数でテストが可能となることを示す。

本論文は次のように構成されている。第2章では、論理回路のテスト方法、概念について簡単に述べる。第3章では、提案するメモリの組込みテスト方式について論じる。第4章では、CMOS組合せ論理回路の開放故障検出のためのテスト容易化設計について、第5章では、組込み電流テストのためのテスト用回路の設計とテスト生成について論じる。第6章では本論文のまとめと今後の課題について述べる。

第 2 章

論理回路のテストの基本概念

論理回路が所定の動作を行わなくなることを故障が発生したといい、この故障を検出する、または、故障箇所を指摘することをテストという。すなわち、テストとは、論理回路の故障の有無を調べることである。本章ではテストについての基本的な概念について述べる。

2.1 節では故障モデルについて述べる。2.2 節のテスト方法では、種々のテストの適用方法について述べる。2.3 節のテスト容易化設計ではその概念と設計方式について述べる。2.4 節メモリのテスト、2.5 節組合せ回路のテストではこれまでに提案されてきたテスト方式、テスト生成法について述べる。2.6 節の電流テストでは新しいテスト方法としての電流テスト法の特徴について述べる。2.7 節で本章のまとめを行う。

2.1 故障モデル

故障は、回路内の信号線の断線、短絡、格子欠陥などによる pn 接合不良などの物理的原因によって発生する。これらの物理的原因によって生じる現象を抽象化したものが故障モデルである。

故障モデルは被テスト回路の記述レベルに応じ、機能レベル、ゲートレベル、トランジスタ・レベルで考えることができる^(7,8)。機能レベルの故障としては、メモリセルの結合故障、パターン依存故障などがある。また、ゲートレベル、および、トランジスタ・レベルの故障としては、それぞれ、信号線の縮退故障、ブリッジ故障、および、トランジスタの開放故障、短絡故障などがある。

また、故障はその捕え方により、論理故障とパラメータ故障、固定故障と間欠故障、単一故障と多重故障などに大別できる⁽¹⁻⁶⁾。論理故障は回路内に故障が存在していても論理回路として動作する故障である。パラメータ故障は動作速度、電圧、電流などの回路パラメータが変動する故障である。固定故障は、一度故障が発生したならば、その影響が永続する故障であり、一方、間欠故障は、一時的に回路内に故障が存在する故障である。また、単一故障は、回路に故障が発生したとき、唯一つの故障しか存在しないものであり、多重故障は同時に複数個の故障が存在するものである。

なお、故障の発生は確率的に表現され、多重故障の発生確率は、単一故障のそれより小さい。また、多くの場合、単一故障を検出するテストベクトルで多重故障をも検出できるといわれている^(1,7,8)。以上のような観点から、本論文では単一故障を仮定する。

2.2 テスト方法

論理回路は、通常、その外部入力端子に適当な入力を印加し、外部出力端子で観測される出力応答を正常な場合のそれと比較することによりテストされる。この入力は、故障の存在時には正常な場合と異なる応答を出力する入力であり、テストベクトルと呼ばれる。また、正常な場合の出力応答は期待値と呼ばれる。図 2.1 に一般的なテスト方法を示す。

テスト方法はその実行形式により多くの方式に分類できる^(5,70)。まず、テストベクトルの生成・処理方法によりオンライン・テスト法とオフライン・テスト法とに大別できる。オンライン・テスト法は被テスト回路の通常動作中にテストを実行する。通常動作中の入

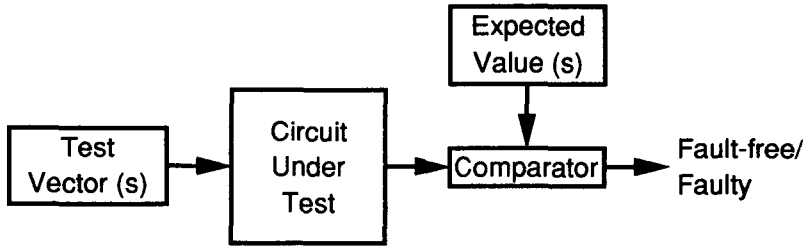


図 2.1 一般的なテスト方法

力がテストベクトルとしての役割を果たす。一方、オフライン・テスト法は被テスト回路の通常動作と分離してテストを行う。この方法にはテストのための特別な入力であるテストベクトルが必要であり、これを被テスト回路に印加し、その出力応答を評価する。オフライン・テスト法はテストの適用方法、テストベクトルの生成方法、出力応答の判定方法により、さらにいくつかに分類できる。

まず、テストの適用方法により外部テスト法と組み込みテスト法とに分類できる。外部テスト法ではテスト装置は被テスト回路外にあり、テストベクトルは外部装置から被テスト回路に印加され、被テスト回路の応答は外部テスト装置へ出力される。一方、組み込みテスト法ではテスト装置が被テスト回路内に組み込まれている。これについては次節 2.3 で述べる。テストベクトルの生成には、与えられた假定故障を検出するための入力を求めるアルゴリズム的手法と、逆に、与えられた入力によって検出可能な故障を求める故障シミュレーション法とがある。アルゴリズム的手法では検出可能なすべての故障に対するテストベクトルを求めることができる。故障シミュレーション法ではすべての可能な入力の組合せについて調べない限り、すべての検出可能な故障に対するテストベクトルが求まるとは限らない。アルゴリズム的手法によるテスト生成には、回路の論理関数を用いてテストベクトルを求める Poage の方法⁽⁷¹⁾、ENF 法⁽⁷²⁾、プール微分法⁽⁷³⁾ などがあるが、現在広く用いられている手法は経路活性化法⁽⁷⁴⁾ という考えに基づいたテスト生成手法である。これについては 2.5 節で述べる。出力応答の判定方法はすべての出力を比較する方法とコンパクトテスト法とに分類できる。前者は期待値、または、既知の正常な回路の出力と被テスト回路の出力とを一括、または、逐次比較し、正常/故障の判定をする方法である。コンパクトテスト法は被テスト回路の出力応答系列を適当な関数で圧縮し、その最終的な値である圧縮値と正常な回路のそれとを比較し、判定する方法である。データ圧縮には 1 計数法、

遷移計数法などがよく知られているが、最近では符号解析法が広く用いられている。符号解析法については次節で述べる。

2.3 テスト容易化設計

通常、ICは外部テスト装置によってテストされる。このテスト方法ではテストベクトルを印加する制御点はICの外部入力端子であり、観測点は外部出力端子である。ICが大規模化、高機能化しても外部入出力端子はそれほど増加しないため、ICの1入出力端子当りの素子数の割合が増加してきている。従って、回路の可制御性・可観測性が低下し、テストが困難になってきており、テストコストの増大を招いている。

この問題に対処するため、現在ではテスト容易化設計と呼ばれる設計手法が用いられている。テスト容易化設計とは、テストを考慮しテストを容易にする回路を設計することであり、回路の可制御性・可観測性を向上させることである。可制御性は、外部入力端子から回路の内部状態を制御できる容易さの尺度であり、可観測性は、外部出力端子で回路の内部状態を観測できる容易さの尺度である⁽⁷⁴⁾。テスト容易化設計の目標はテスト生成時間の短縮、テストベクトル数の削減である。テスト容易化設計では回路の機能を実現するために必要な素子の他に、テストのための回路が余分に必要である。従って、テスト容易化設計に対する評価項目としては、目標の達成度の他に、付加面積オーバーヘッド、動作速度の遅れなどがある。

テスト容易化設計の手法は、アドホック設計 (Ad Hoc Design) と構造的設計 (Structured Design) とに大別できる^(5,8,9)。アドホック設計は個々の回路のテストの困難さの問題を解決する設計手法のことである。回路を適当な大きさに分割してテスト生成時間の減少、可観測性・可制御性の向上をはかったり、あるいは、回路の内部状態の制御・観測を容易にするための付加入出力端子を設ける方法などがある。一方、構造的設計は、テストを容易にする設計方針を用いてテストの困難さに関する問題を解決する設計手法である。その代表的なものとして、順序回路に対するスキャン設計法⁽⁷⁵⁻⁷⁷⁾が知られている。これは図2.2に示したように順序回路の内部状態を直接制御・観測できるようにスキャンイン、スキャンアウトの付加入出力端子を設け、かつ、記憶素子としてテスト時にシフトレジスタの動作が可能なフリップ・フロップを使用した設計である。

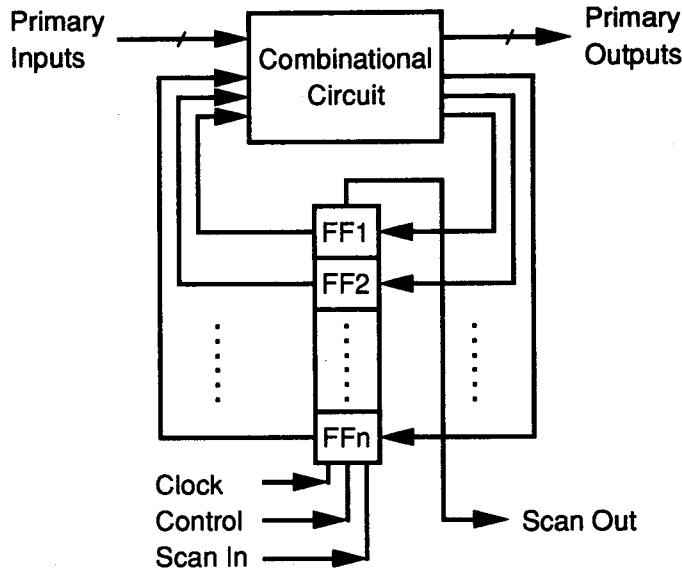
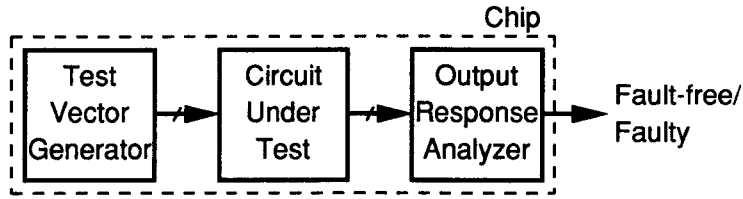
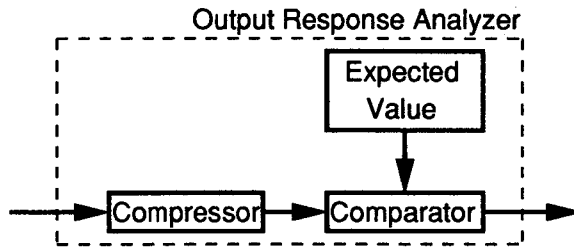


図 2.2 スキャン設計法

組込みテスト法は外部テスト法に対抗するテスト方法であるが、外部テスト装置の負担を軽減するという意味においてテスト容易化設計の一手法である。組込みテスト法は図 2.3 (a) に示すように、テストベクトル生成部と出力判定部とから成るテスト機構を被テスト回路と同一のチップ内に組込むことで実現される。これらのテスト機構は、付加面積オーバーヘッドを小さくするために被テスト回路に比べ簡単に実現されなければならない。テストベクトルとしては全数テストベクトル、擬似乱数テストベクトル、マイクロコードによるテストベクトルなどが使用される。擬似乱数は線形フィードバック・シフトレジスタ (Linear Feedback Shift Register : LFSR) によって容易に実現可能なので広く用いられている。出力判定には、期待値の記憶容量を減らすためにコンパクトテスト法 (図 2.3 (b)) が用いられる。データ圧縮には LFSR により実現される符号解析法^(78, 79)を用いるのが一般的である。4 ビットの LFSR による符号解析器の回路を図 2.4 に示す。符号解析法ではレジスタ内に最後に残った値が圧縮値であり、シグネチャ (Signature) と呼ばれる。しかし、この方法の問題点として、データ圧縮による故障の見逃しであるエイリアス誤り (Aliasing Error)^(80, 81)が存在する。エイリアス誤りとは、符号解析器への入力データ内に故障による誤りが存在していても、データ圧縮によってその誤り情報が失われ、正常な場合のシグネチャと等しくなり、故障を見逃してしまうことである。LFSR のシフトレジスタのビット数を m ,



(a) 組込みテスト法の回路構成



(b) コンパクトテスト法の回路構成

図 2.3 組込みテスト法とコンパクトテスト法

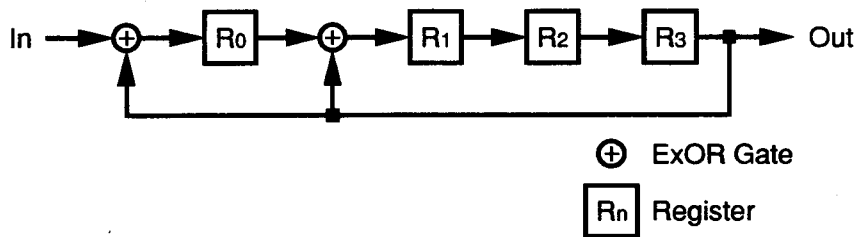


図 2.4 4 ビット線形フィードバック・シフトレジスタ

LFSR への入力データの長さを n ビット，エイリアス誤りの発生確率を P とし，すべての可能な入力データの出現確率が等しいとした場合，

$$P = (2^{n \cdot m} - 1) / (2^n - 1) \tag{2.1}$$

である⁽⁷⁾。

$$n \leq m$$

ならば

$$P = 0,$$

$$n \gg m$$

ならば

$$P = 2^{-m}$$

である (79, 80).

2.4 メモリのテスト

メモリのテストは、その機能、構造から一般の論理回路とは異なる方法で行われる。メモリの機能は、他のセルのいかなる状態に対しても、任意のセルに任意のデータを正しく書込むことができ、そのデータを保持し、かつ、読出すことができることである。従って、メモリのテストは、データの書込み・読出し操作、すなわち、セルへのアクセスによって行われる。テスト時間はテストに必要なアクセス回数の総和、すなわちテスト系列長に比例する。

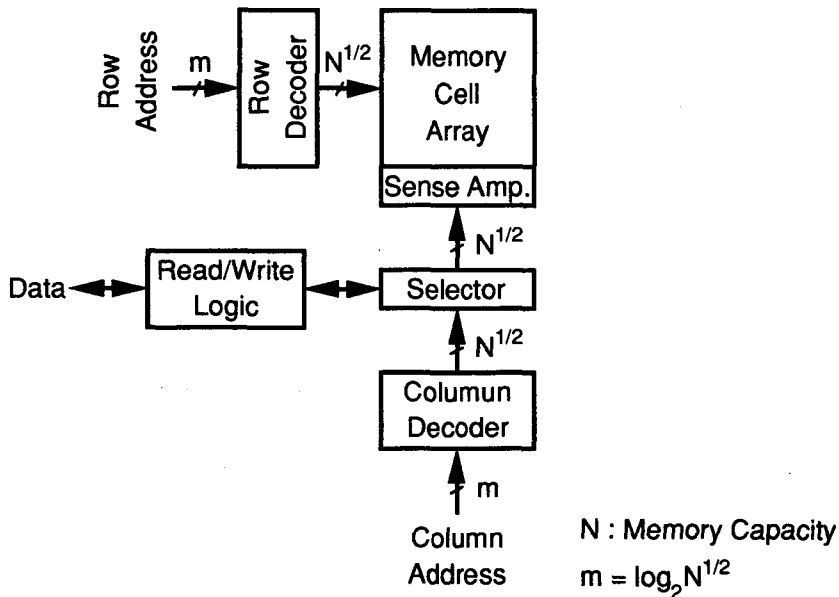


図 2.5 メモリの基本構成

メモリは図 2.5 に示すように、主にメモリ・セルアレイとデコーダとから成っている。セルアレイの故障⁽³³⁾としては、セルの内容が '0' または '1' のどちらか一方に固定される縮退故障、複数のセルが電氣的に結合し、これらの中のあるセルの内容の変化が結合している他のセルの内容を変化させる結合故障 (Coupling Fault)、あるセルの内容が他の複数のセルのパターンにより影響を受けるパターン依存故障 (Pattern-Sensitive Fault) などがある。結合故障やパターン依存故障の影響の範囲によってテスト系列の長さが異なる。

[例 2.1] テスト系列長が最も短いものの一つであるマーチングテスト (Marching 1's and 0's)^(1, 33) の手続きを図 2.6 (a) に示す。アドレス i のセルにデータ d を書込むことを $W_i(d)$ 、アドレス j のセルの内容を読出すことを R_j と表している。この手続きでは、初めにセルの内容を初期化し (図 2.6 (a) の (1))、次に各セルの内容の読出し、データの遷移、書込んだデータの読出しをアドレスの昇順 (図 2.6 (a) の (2))、降順 (図 2.6 (a) の (3)) に行う。これらの操作を '0'、'1' それぞれに対して行う。 $d=0$ の場合の操作手順を図 2.6 (b) に示す。1 ビット/ワードの記憶容量 N ビットのメモリにこのテストを外部テスト装置を用いて適用した場合、テスト系列長は $14N$ である。なお、マーチングテストではセルの縮退故障と隣接セル間の一部の結合故障とを検出できる。

通常、同一行内、または、同一列内の結合故障、パターン依存故障を検出するためには $O(N^2)$ のテストが必要であり、セルアレイ内の任意のセル間でのそれらを検出するためには $O(N^2)$ のテストが必要である^(3, 4, 33)。

正常なデコーダは入力と出力とが全単射であるため、これを保証するテストを行えばデコーダの故障に対するテストを行うことができる。デコーダは図 2.7 に示す長さ $5N$ のテストでテスト可能である⁽²⁷⁾。このテストにおいて、図 2.7 中に示した (1) と (2) の操作は図 2.6 (a) に示したマーチングテストの中にも含まれているので、マーチングテストはデコーダに対するテストも行っている。

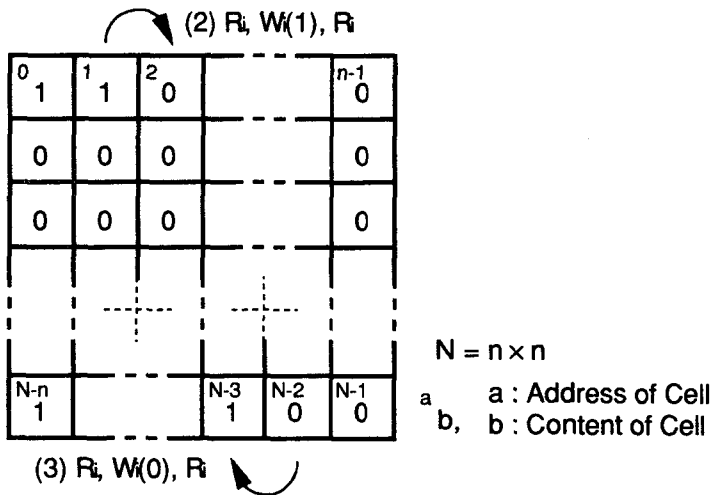
2.5 組合せ回路のテスト

論理回路はテストベクトルを印加し、その出力応答を観測することでテストされる。従って、論理回路のテストにおける最大の問題は、効率よくテストベクトルを求めるといふテスト生成に関するものであり、これまでに多くの研究がなされている⁽¹⁻⁸⁾。本節では外部テスト法による組合せ回路のテスト生成手法について述べる。

```

procedure Marching 1's and 0's Test
  /* Test procedure for cell array */
  begin
    for d = 0 to 1 do begin
      for i = 0 to N-1
         $W_i(d);$  ..... (1)
      for i = 0 to N-1
         $R_i, W_i(\overline{d}), R_i;$  ..... (2)
      for i = N-1 to 0
         $R_i, W_i(d), R_i;$  ..... (3)
    end
  end;
  
```

(a) 手続き



(b) 操作手順

図 2.6 マーチングテスト

```

procedure Nair, Thatte and Abraham's Method
  /* Test procedure for decoder */
  begin
    for i = 0 to N-1
       $W_i(0);$ 
    for i = 0 to N-1
       $R_i, W_i(1);$  ..... (1)
    for i = N-1 to 0
       $R_i, W_i(0);$  ..... (2)
  end;
  
```

図 2.7 デコーダのテスト手続き

初めに用語の定義を行う。

[定義 2.1] 回路内の故障の有無を区別できる外部入力値をテストベクトルと呼ぶ。

[定義 2.2] テストベクトルをある特定の順番に回路に印加しなければならないとき、その並びをも考慮したテストベクトルの集合をテスト系列と呼ぶ。

[定義 2.3] 回路内に故障の影響が現れるような入力を印加することを故障の顕在化 (Fault Excitation) と呼ぶ。また、このとき故障は顕在化されたという。

[定義 2.4] 故障の影響を故障位置から回路の外部出力に伝搬させることを故障の影響の伝搬 (Fault Effect Propagation) と呼ぶ。

[定義 2.5] 故障の影響を伝搬させるために、回路の外部入力値を決定することを信号線の正当化 (Line Justification)、または、後方操作 (Backward Trace) と呼ぶ。

n 入力 $X = (x_1, x_2, \dots, x_n)$ の組合せ回路の出力関数を $f(X)$ とし、ある故障 i により出力関数が $f_i(X)$ になったとする。外部テスト法においては、 $f(X) \oplus f_i(X) = 1$ を満足する入力 X が故障 i に対するテストベクトルである。

この様なテストベクトルを求める手法には、回路の論理関数を用いることもできるが、回路規模が大きくなるとテスト生成に必要な計算時間や記憶容量が膨大になり実用化が困難である。現在では経路活性化法と呼ばれる手法が多数研究されており、また、実用化されている。

まず経路活性化法に基づいたテスト生成手法について紹介する。経路活性化法では、通常、信号線の論理値が '0' または '1' に固定される縮退故障を対象としている。経路活性化法の基本原理は、故障の存在する位置から外部出力まで、故障の影響を伝搬させる経路を求めることである。このとき、故障の影響を伝搬した経路を活性化経路と呼ぶ。その基本操作は、故障の顕在化、故障の影響の伝搬、信号線の正当化である。

[例 2.2] 故障の顕在化は故障位置に正常時と故障時とで異なる値を設定し、これを満足する入力値を求める操作である (図 2.8 の領域 (1))。すなわち、図 2.8 の回路において、縮退故障 SA- α の発生箇所に論理値 $\bar{\alpha}$ を設定する入力値を求めることである。また、故障の影響の伝搬は、故障の顕在化によって設定された値を外部出力まで伝搬できるように活性化経路を選択し、その経路上の各ゲート G_i ($i = 1, 2, \dots, n$) の入力値を設定する操作である (図 2.8 の領域 (2))。さらに、信号線の正当化は、活性化経路上の各ゲート G_i の入力に設定した値になるように回路の外部入力値を決定する操作である (図 2.8 の領域 (3))。

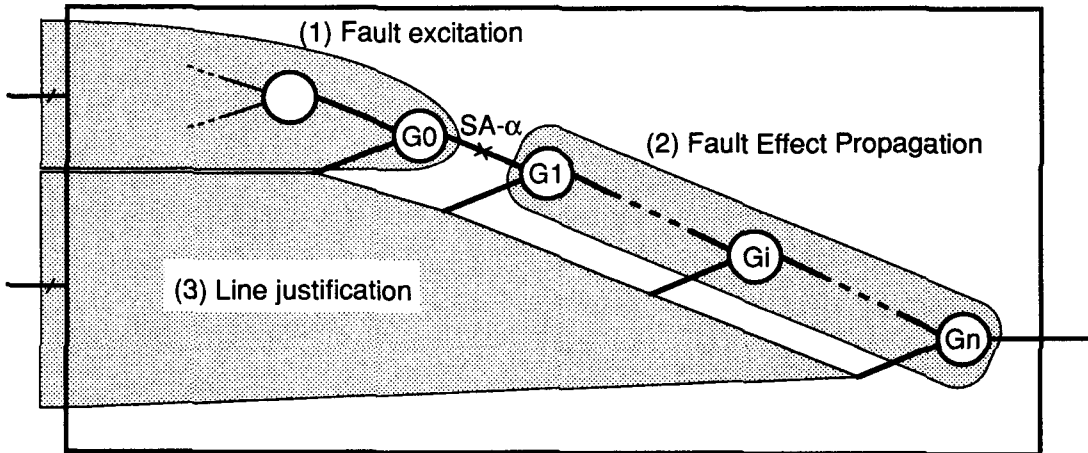


図 2.8 経路活性化法

経路活性化法の代表的なテスト生成手法である D アルゴリズム⁽⁸²⁾と PODEM アルゴリズム⁽⁸³⁾ について説明する。

[例 2.3] 図 2.9 に示す回路を用いて D アルゴリズムによるテスト生成について説明する。信号線 e の 1 縮退故障 SA-1 を考える。この故障を顕在化するためには e の値が '0' になるような入力を印加しなければならない。よって $b = c = 1$ となる。故障を顕在化するための操作を D 設定操作と呼ぶ。正常時 '1' ('0'), 故障時 '0' ('1') となる値を記号 $D(\overline{D})$ で表す。 $f = \overline{D}$ であることから一意的に $f = g = \overline{D}$, $i = D$ と決まる。この様に、ある信号線の値により一意的に決定される信号線の値を求める操作を含意操作と呼ぶ。次に故障の影響 $f = \overline{D}$ をゲート G3 の出力に伝搬させるために $a = 1$ とし、よって $j = D$ となる。故障の影響 D あるいは \overline{D} を外部出力側へ伝搬させるための操作を D ドライブといい、この操作を繰り返して故障の影響の伝搬を行うことを D 伝搬という。同様に、ゲート G6 に対しても D ドライブを行うと $k = 0$, $l = \overline{D}$ となり、外部出力に故障の影響 \overline{D} が伝搬する。最後に信号線 k の正当化を行う。D アルゴリズムにおいて、信号線の正当化は一致操作と呼ばれる。ゲート G5 において、 $i = D$ となっているので一致操作により $h = 1$ となる。またゲート G2 についても一致操作により $d = 0$ となる。以上の操作から e の 1 縮退故障に対するテストベクトルは $(x_1, x_2, x_3, x_4) = (1, 1, 1, 0)$ として求まる。

[例 2.4] PODEM アルゴリズムによるテスト生成について説明する。D アルゴリズムでは故障の影響の伝搬を行ってから信号線の正当化を行い、外部入力値を求めていたのに

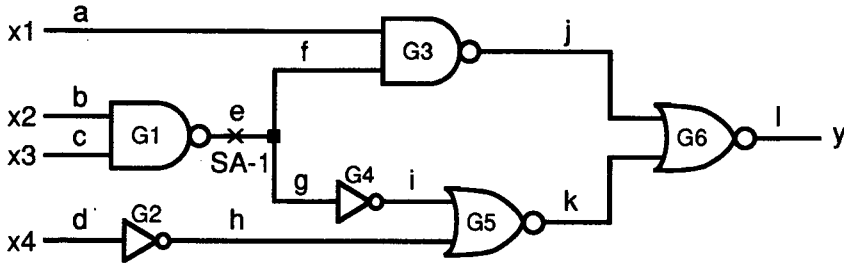


図 2.9 回路例 1

対し、PODEM アルゴリズムでは外部入力に値を割当て、含意操作で故障の影響の伝搬を行う。D アルゴリズムと同様に図 2.9 の信号線 e の 1 縮退故障 SA-1 について考える。まず D 設定操作により $b = c = 1$ となり、含意操作により $f = g = \overline{D}$, $i = D$ となる。 $f = \overline{D}$ をゲート G3 の出力に伝搬させるために $a = 1$ となる。次にゲート G6 について D ドライブを行う。このとき $k = 0$ とする必要がある。よって、信号線 k に '0' を設定するために、信号線の正当化を行わなければならない。PODEM アルゴリズムでは、入力側に向かって 1 本の経路を選び、到達した外部入力に値を割当てるという後方追跡を用いて信号線の正当化を行う。後方追跡により $d = 0$ となる。 $d = 0$ について含意操作を行い $h = 1$ となり、 $k = 0$ という目標値が得られる。従って、 \overline{D} が外部出力に伝搬したので $(x1, x2, x3, x4) = (1, 1, 1, 0)$ がテストベクトルとなる。

[定義 2.6] どのようなテストベクトルを回路に印加しても、その出力応答が等しく、互いに区別できない故障を等価故障と呼ぶ。また、等価故障の中から選ばれた 1 個の故障を代表故障と呼ぶ。

[例 2.5] 図 2.9 の回路において、信号線 e の 1 縮退故障に対するテストベクトル $(x1, x2, x3, x4) = (1, 1, 1, 0)$ は、信号線 f, l の 1 縮退故障、信号線 b, c, j の 0 縮退故障をもテストでき、また、出力信号線は 1 本であるため、これらの故障を区別できない。よって、これらの故障は e の 1 縮退故障と等価故障である。さらに、この中の 1 個の故障、例えば、e の 1 縮退故障が代表故障である。

なお、経路活性化法では、故障の影響の伝搬、信号線の正当化の操作を行うとき、いくつかの選択肢が存在する。ある時点で信号値の設定に矛盾が生じた場合、一つ手前の選択点に戻り別のものを選択しなければならない。このような操作をバックトラックと呼ぶ。バックトラックが頻繁に発生するとテスト生成時間が長くなるため、実際のテスト生成で

は、バックトラックの回数に上限値を定めており、この上限値を越えたとき、その故障に対するテスト生成を中止する。

次に、CMOS 組合せ回路の MOS トランジスタの開放故障を検出するためのテスト手法について説明する。開放故障とは、MOS トランジスタのゲート電圧にかかわらず、常にトランジスタがオフ状態となる故障である。完全相補形 CMOS ゲートは、図 2.10 に示すように pMOS トランジスタ回路と nMOS トランジスタ回路とで構成されている。それぞれの回路構成は、一方の回路内のトランジスタが直列に接続されていれば、他方の同一入力信号線に接続されたトランジスタは並列に接続されており、互いに論理的に双対な回路である。すなわち、回路が正常な場合、任意の入力に対し V_{DD} 、または、GND から出力へのどちらか一方のみの導通経路が形成され、出力はそれぞれ '1'、または、'0' となる。開放故障の存在のため、ある入力に対し V_{DD} 、および、GND と出力との間に導通経路が形成されない場合、すなわち、故障が顕在化されたとき、出力はハイ・インピーダンス状態になる。このとき、出力は負荷容量 C_L の存在により、ハイ・インピーダンス状態になる直前に C_L 内に充放電された電圧を保持する。すなわち、開放故障の存在により、ある入力に対し、組合せ回路は順序回路の動作をする。なお、負荷容量 C_L は配線容量、ゲート容量、寄生容量などから成っている。

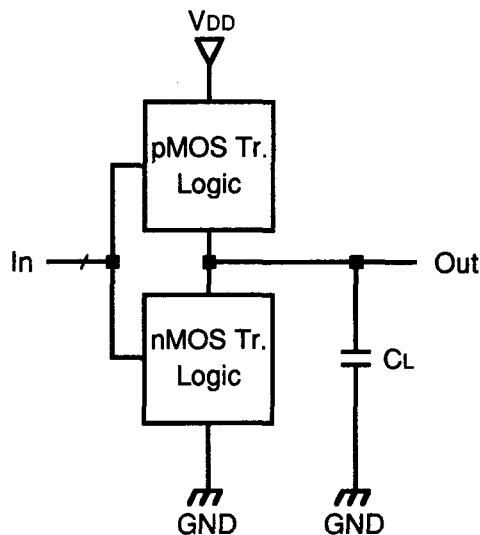


図 2.10 完全相補形 CMOS ゲート

【定義 2.7】 1 個の開放故障を検出するために、連続した 2 つのテストベクトル $\langle T_1, T_2 \rangle$ の印加を必要とするテストを 2 パターンテスト⁽⁴⁹⁾と呼ぶ。また、この 2 つのテストベクトルのうち、第 1 のベクトル T_1 を初期化ベクトル、第 2 のベクトル T_2 を検出ベクトルと呼ぶ。

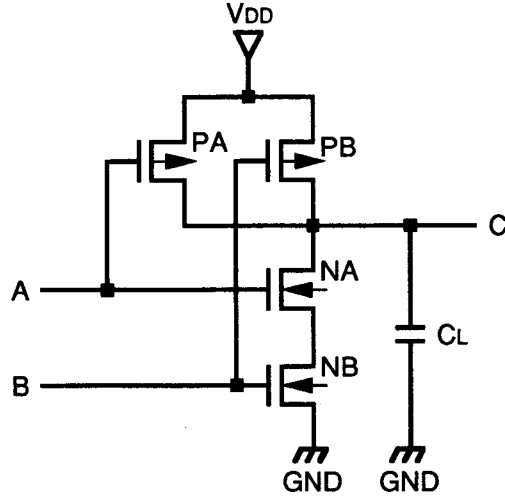
【定義 2.8】 初期化ベクトルによるゲートの出力値を初期値と呼ぶ。

2 パターンテストにおいて、初期化ベクトルは、検出ベクトルを印加したとき故障ゲートの出力を正常なゲート出力と補数となる論理値に初期化する。検出ベクトルは、故障ゲートの出力をハイ・インピーダンス状態にし、かつ、故障の影響、すなわち、初期化ベクトルによって設定された論理値を外部出力まで伝搬する。

【定義 2.9】 タイミング・スキューや回路遅延などの存在にかかわらず、開放故障の検出を正常に行えるテストをローバストテスト^(51,52)と呼ぶ。また、このときのテストベクトルをローバスト・テストベクトルと呼ぶ。

【例 2.6】 図 2.11 (a) の 2 入力 NAND ゲートについて考える。このゲート内の各トランジスタが正常な場合と故障の場合との真理値表を図 2.11 (b) に示す。この図中で、最後の 3 つの欄は、トランジスタ PA, PB, NA と NB, それぞれに開放故障が存在する場合の出力値を表している。Z はハイ・インピーダンス状態を表す。出力がハイ・インピーダンス状態となり故障が顕在化されたとき、これを検出するためには、正常時と故障時とで異なる論理値を出力させなければならない。このため、図 2.11 (c) に示した初期化ベクトル T_1 を印加し、続いて検出ベクトル T_2 を印加しなければならない。 T_2 を印加したとき、各トランジスタが正常ならば図 2.11 (b) に示した正常時の値を出力し、開放故障が存在すれば図 2.11 (c) の最後の欄に示した値を出力する。なお、nMOS トランジスタ NA と NB のように、同一ゲート内で直列に接続されたトランジスタの開放故障は、同一のテストベクトルで検出され互いに区別できない等価故障である。

上述したように、開放故障に対する 2 パターンテストでは、開放故障の顕在化時における負荷容量内の充放電された電圧を利用しているので、 T_1 から T_2 へ入力に変化するとき、外部入力端子でのタイミング・スキューや回路遅延のために T_1 と T_2 との間で不正な論理値が発生し、負荷容量内に保持されていた初期値が破壊されテストが無効、すなわち、故障検出が不可能となる場合もあり得る。



(a) 2入力CMOS NAND ゲート

A	B	C			
		Fault-free	PA	PB	NA, NB
0	1	1	Z	1	1
1	0	1	1	Z	1
1	1	0	0	0	Z

Z : High-Impedance State

(b) 真理値表

Faulty Tr.	T ₁ (A, B)	T ₂ (A, B)	C
PA	(1, 1)	(0, 1)	0
PB	(1, 1)	(1, 0)	0
NA, NB	(0, ∅), (∅, 0)	(1, 1)	1

$\emptyset = \{0, 1\}$

(c) テストベクトル

図 2.11 開放故障のテストベクトル

[例 2.7] 図 2.12 (a) の回路について 2 パターンテストが無効となる場合について考えてみる。いま、ゲート G3 内の pMOS トランジスタ PF の開放故障について考える (図 2.12 (b))。テストベクトルとして $\langle T_1(A, B, C, D), T_2(A, B, C, D) \rangle = \langle (0, 1, 0, 1), (1, 1, 1, 0) \rangle$ を選んだとする。ここで、 T_1 から T_2 への入力遷移が行われたとき、タイミング・スキューや回路遅延のためゲート G2 の入力 E が入力 C より遅れてデータ遷移したならば、ハザード $T_h(A, B, C, D) = (1, 1, 1, 1)$ がテストベクトルの遷移間に発生する。この T_h はトランジスタ

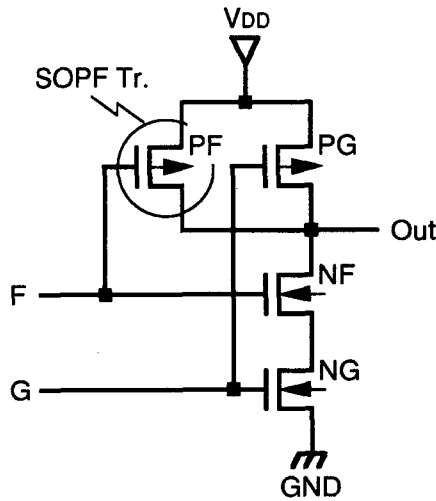
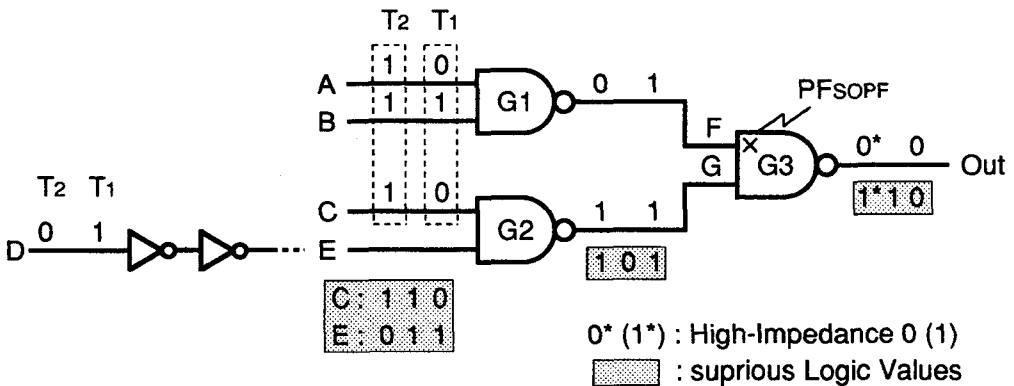


図 2.12 回路例 2

タ PG を導通させるので G3 の出力論理値は不正に '1' になり, T_1 によって設定された初期値が破壊される. 続いて安定な入力 T_2 が印加され, G3 の出力をハイ・インピーダンス状態にする. このとき出力論理値はトランジスタ PF が故障であるにもかかわらず '1' であり, T_2 印加時の正常なゲート出力の期待値と同一である. 従って, テスト $\langle T_1, T_2 \rangle = \langle (0, 1, 0, 1), (1, 1, 1, 0) \rangle$ はトランジスタ PF の開放故障の検出に失敗する可能性がある.

このようなタイミング・スキューや回路遅延による 2 パターンテストの潜在的な無効を避けるテストとして, ローバストテストが提案されている. 通常, ローバストテストでは 2 パターンテストの T_1 と T_2 との間で, ハミング距離が 1 であるテストベクトルが使用される. 先の図 2.12 (a) の回路例においては, $T_1(A, B, C, D) = (0, 1, 0, 1)$ のとき, T_2 となり得る候補として $\{(A, B, C, D)\} = \{(1, 1, 1, 0), (1, 1, 0, 0), (1, 1, 0, 1)\}$ があるが, この中から $T_2(A, B, C, D) = (1, 1, 0, 1)$ を選ぶことによりローバスト・テストベクトルを得ることができる.

2.6 電流テスト

これまで説明してきたテストは IC の出力応答を観測して故障を検出するものであった. 換言すれば, 回路の出力電圧に基づいたテストである. これに対し, 最近, CMOS IC の電源電流を計測して故障検出を行う電流テスト, または, I_{DDQ} テストが提案されている. この背景として, CMOS 回路に固有なトランジスタの短絡故障, 開放故障, ゲート酸化膜短絡故障, pn 接合不良などの縮退故障では説明できない故障が現れるようになってきたことが挙げられる. これらの故障は, その影響がゲート出力に現れることが少ないため, 前節で述べた縮退故障を対象としたテストではその検出を保証できない. 正常な CMOS IC では回路動作が安定状態にあるとき静的電流がほとんど流れないのに対し, 縮退故障を含めて, 上記した故障が顕在化された場合には異常な静的電流が流れるという特徴がある. この特徴に着目したテスト方法が電流テスト法である. 図 2.13 に電流テスト法の構成を示す. CMOS IC の電流計測は DC テスト項目の一つである静消費電流として従来から行われてきている. しかし, 本節で述べる電流テストは故障検出のためにテストベクトルを印加し, 各々のテストベクトルについて電流値を計測するという点で DC テストとは本質的に異なる.

図 2.14 に示した CMOS インバータを用いて基本的な DC 特性と故障時における現象とについて説明する. MOS トランジスタの V-I 特性方程式を pMOS トランジスタと nMOS

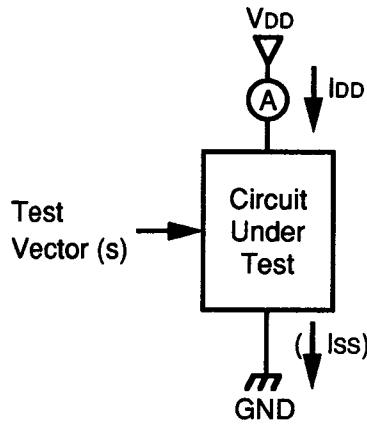


図 2.13 電流テスト法

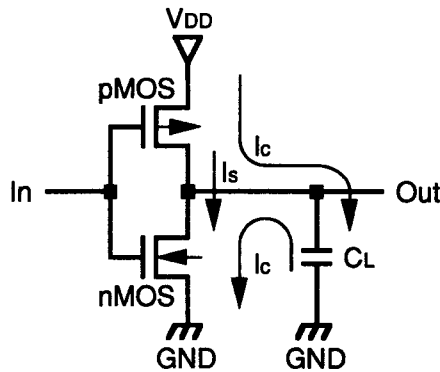


図 2.14 CMOS インバータ

トランジスタとについて解くと図 2.15 に示す CMOS インバータの DC 特性を得ることができる。ここで I_{dsn} (I_{dsp}) は nMOS (pMOS) トランジスタのドレイン-ソース電流であり、この場合 I_{DD} (I_{SS}) と同一である。また、 V_n (V_p) は nMOS (pMOS) トランジスタのしきい値電圧である。完全相補形 CMOS 回路では、このようにゲート端子への印加電圧 V_n が中間電圧 ($V_n < V_n < V_{DD} - IV_p$) にあるときは動的電流が流れるが、入力が安定しているときは静的電流がほとんど流れない。なお、動的電流には、図 2.14 に示すようにゲートのスイッチングによる貫通電流 I_s と負荷容量 C_L による充放電電流 I_c とがある。

次に示す例で CMOS 回路の導通故障時の現象について説明する。

[例 2.8] 図 2.14 の回路の pMOS トランジスタに導通故障が発生した場合を考える。

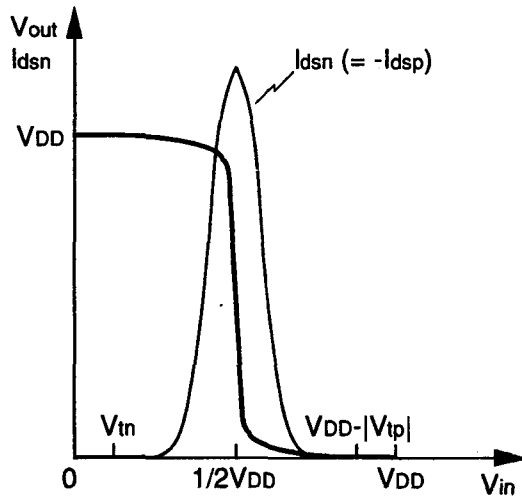


図 2.15 CMOS インバータの DC 特性

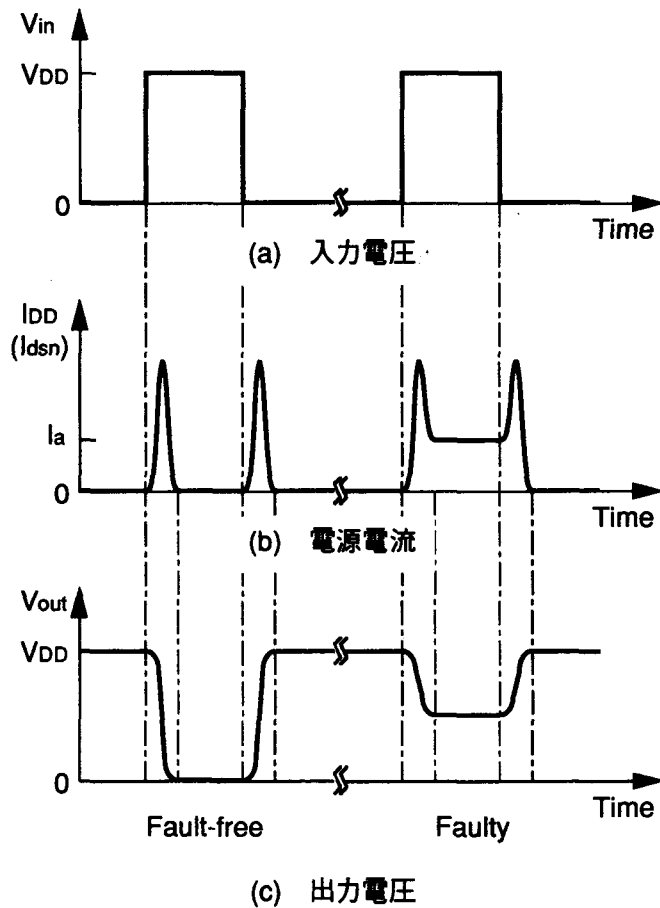


図 2.16 CMOS インバータの電源電流の時間波形

図2.16(a)の入力を印加したとき、入力が論理値'0'のときは正常な場合と同様の動作をするが、入力が論理値'1'のときはpMOSトランジスタとnMOSトランジスタとの両方が導通し、異常な静的電流 I_{q} が流れる(図2.16(b))。よって、この I_{q} を計測することで故障検出が可能となる。ここで、 I_{q} は故障時における貫通電流である。なお、CMOSゲート内の導通故障が顕在化したとき、その出力電圧は中間電圧となる(図2.16(c))。この電圧値は、電圧源と出力との間で導通しているnMOSトランジスタのオン抵抗値とpMOSトランジスタのそれとの比によって決まる。通常のCMOSゲートでは、ゲートのしきい値電圧が $1/2V_{\text{DD}}$ で、かつ、立上り時間、立下り時間を速くする設計をしているため、CMOSインバータに導通故障が起き、かつ、故障トランジスタのオン抵抗が正常時のそれと同一であれば、出力電圧はほぼ $1/2V_{\text{DD}}$ となり、論理値としては不定である。従って、ゲート出力の論理値により故障の検出を行う縮退故障のテストではテスト不可能である。

以上、MOSトランジスタの導通故障を例にして故障の影響について説明したが、CMOS回路内になんらかの故障が発生した場合、多くの場合異常な静的電流が流れ、その存在を知ることができる。

2.7 総括

本章では、本論文で必要とする論理回路のテストに関する基本概念、テスト手法について述べた。まず、故障モデル、および、テストの適用方法によるテスト方法の分類を述べた後、メモリに対する外部テストによるテスト方法について述べた。また、組合せ回路の縮退故障に対する代表的なテスト生成手法とトランジスタの開放故障に対するテスト方法について述べた。さらに、電流テスト法の特徴とトランジスタの導通故障の現象について述べた。

第 3 章

メモリの組込みテスト

メモリは、主にセルアレイ、デコーダ、読出し・書込み回路から成る。本章では、これらに発生する故障を対象とした組込みテストについて論じる。特に、セルアレイについては、その検出が最も困難であると考えられているパターン依存故障を対象とする。メモリのテストでは、テストベクトルの印加はデータの書込みに、出力応答はセルの内容の読出しに対応する。従って、メモリのテストは読出し・書込み操作の繰返しによって行われる。

まず、3.1 節で故障モデルの定義と検出条件について述べた後、3.2 節で提案するメモリの組込みテスト方式について述べる。3.3 節では、提案するテスト方式に適したテスト、すなわち、テスト手続きについて述べ、3.4 節、3.5 節では、出力応答の判定、テスト用回路についてそれぞれ述べる。3.6 節では本テスト方式の評価をし、3.7 節で本章のまとめを行う。

本論文で対象とするメモリは、記憶容量 N の正方アレイ ($N = N^{1/2} \times N^{1/2}$) で、かつ、1 ビット/ワードの SRAM であると仮定する。

3.1 故障モデルと検出条件

故障による物理的現象を抽象化したものが故障モデルである。故障モデルは被テスト回路の記述レベルに応じ、機能レベル、ゲートレベル、トランジスタ・レベルで考えることができるが、メモリに対しては、その機能を考慮してテストした方がテストし易い。ここではメモリの機能レベルの故障を定義する。なお、故障は単一固定故障のみを考える。

図3.1に本論文で考えるメモリモデルを示す。メモリはセルアレイ、デコーダ、読出し・書込み回路に大別することができる。

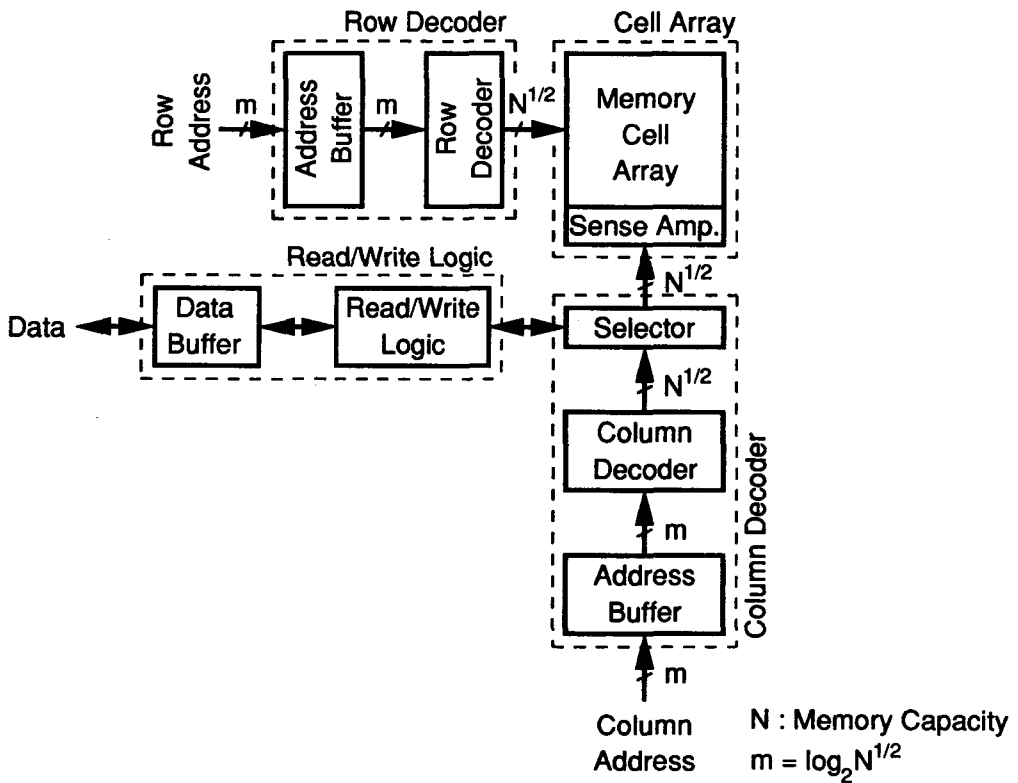


図3.1 メモリモデル

3.1.1 故障モデル

A. セルアレイの故障

[定義 3.1] あるセルの内容が、他のセルの内容にかかわらず '0' あるいは '1' に固定

する故障をセル縮退故障 (Cell Stuck-At Fault) と呼ぶ。

[定義 3.2] 電氣的に結合されている 2 個以上のセルが存在し、これによって起こる故障を結合故障 (Coupling Fault) と呼ぶ。結合故障には、結合している他のセルの内容のため結合されたセルの内容が変化しない静的結合故障 (Static Coupling Fault) と、結合している他のセルの '0' ('1') から '1' ('0') への書込み操作によるデータ遷移のため結合されたセルの内容が変化しない動的結合故障 (Dynamic Coupling Fault) とがある。但し、残りのセルの内容に無関係である。

[定義 3.3] あるセルが、他の複数のセルのパターンにより影響を受ける故障をパターン依存故障 (Pattern-Sensitive Fault) と呼ぶ。パターン依存故障は故障が発生したときの状況により次の 3 種類の故障に分類できる。他のセルのあるパターンのため変化しない静的パターン依存故障 (Static Pattern-Sensitive Fault)、他の 1 個のセルの '0' ('1') から '1' ('0') への書込み操作によるデータ遷移と残りのセルのパターンとのため変化しない動的パターン依存故障 (Dynamic Pattern-Sensitive Fault)、他のセルのあるパターンに影響され書込み操作でセルの内容の書換えができない遷移パターン依存故障 (Transition Pattern-Sensitive Fault) である。

パターン依存故障に関連した用語の定義を以下に与える。

[定義 3.4] アレイ中の任意の 1 個のセルを基底セル (Base Cell)、パターン依存故障の影響を受ける基底セルとパターン依存故障の影響を与えるセルとの集合を近傍セル (Neighboring Cell) と呼ぶ。また、近傍セルから基底セルを除いた残りのセルを隣接近傍セル (Adjacent Neighboring Cell) と呼ぶ。

結合故障、パターン依存故障は、漏れ電流や寄生容量結合などにより起こるものと考えられている^(27,31,32)。また、セルアレイの構成とその動作を考慮することにより、これらの故障に関連しいくつかの仮定を得ることができる。

[仮定 3.1] 本論文では SRAM、すなわち、非破壊読出しのメモリを対象としているため読出し操作、無遷移書込み操作では結合故障、パターン依存故障は起こらないものとする。また、結合故障は 2 個のセル間でしか起こらないものとする。隣接近傍セルを基底セルの上下左右 4 個のセルとする。なお、基底セルおよびその上下左右の隣接近傍セルをそれぞれ B, W, X, Y, Z とラベル付けする (図 3.2)。

なお、同一近傍セル内では、結合故障はパターン依存故障と等価な故障である。

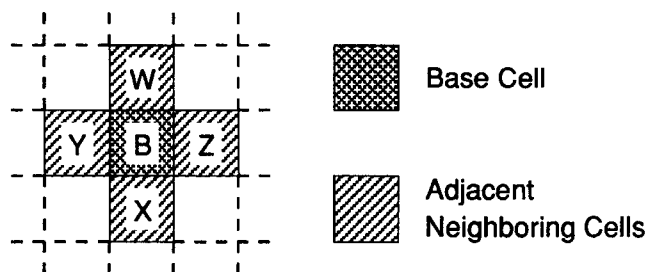


図 3.2 近傍セル

B. デコーダの故障

正常なデコーダはその入力と出力との対応が全単射である。よって、デコーダの機能から故障として次のものを考えることができる。

[定義 3.5] デコーダの入出力の対応が全単射にならないときデコーダは故障であるといい、この故障をデコーダの故障 (Decoder Fault) と呼ぶ。デコーダの故障には、デコーダにアドレスを入力したとき、どのセルも選択されない無選択故障 (Non-Access Fault)、指定したアドレスと異なるアドレスに対応するセルが選択される誤選択故障 (Incorrect Access Fault)、複数のセルが同時に選択される多重選択故障 (Multiple-Access Fault) がある。

論理アドレスとセルの物理的位置との対応を表す AD グラフ (Address Decoder Graph)^(37, 38, 40) を用いてデコーダが正常な場合の対応と故障の場合のそれとを図 3.3 に示す。このグラフから、デコーダが正常な場合入出力の対応が全単射であり、これ以外の対応のとき故障であることがわかる。なお、多重選択故障は図 3.3 (d) の入出力の対応から明らかのように単一故障で存在せず、無選択故障、誤選択故障いずれか一方とあるいは両方と同時に存在する多重故障である。

デコーダの故障の検出問題を簡単にするため、デコーダの故障に対しいくつかの仮定を以下に与える。

[仮定 3.2] 無選択故障が存在する論理アドレスをデコーダに入力したとき、読出されるデータは '0' か '1' に固定されるものとする。また、多重選択故障が存在する場合、書込み操作では選択されたすべてのセルに書込みが行われ、読出し操作では選択されたすべてのセルの内容の AND か OR のいずれかで出力されるものとする。AND になるか OR になるかはそのメモリによって決まるものとする。さらに、誤選択故障が存在する場合、書込

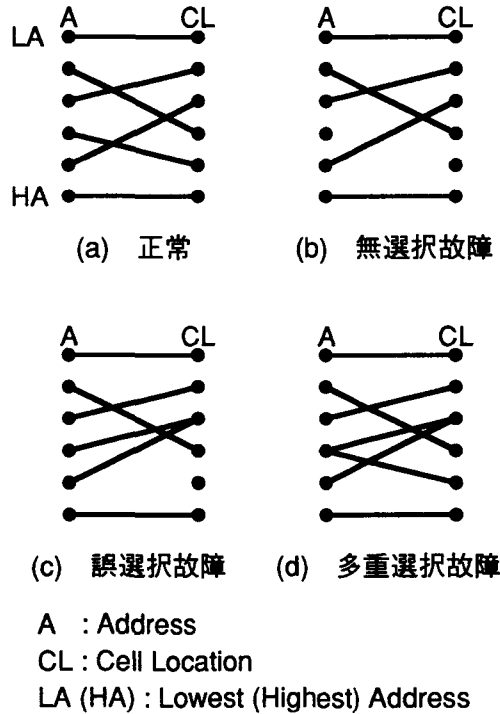


図 3.3 AD グラフ

み操作では最後に書込んだデータがセルに保持され、読出し操作ではすべて同一の内容が出力されるものとする。

C. 読出し・書込み回路の故障

[定義3.6] 読出しするセルの内容、書込みデータに無関係に一定の値'0'または'1'のみ読出し、書込みが行われる故障を読出し・書込み回路縮退故障 (Read/Write Logic Stuck-At Fault) という。

D. その他のメモリ構成回路の故障

メモリ構成回路のその他の部分の故障として次のものが考えられる。

- (1) センスアンプの動作不良故障
- (2) データ入出力線の縮退故障
- (3) データバッファの縮退故障

- (4) セレクタの動作不良故障
- (5) アドレス線の縮退故障，線間短絡故障
- (6) アドレスバッファの縮退故障，結合故障

これらの故障はセルアレイ，デコーダ，読出し・書込み回路のいずれかの故障と等価である。従って，以下，本論文ではこれらセルアレイ，デコーダ，読出し・書込み回路の故障検出のみについて考えることにする。

3.1.2 検出条件

メモリの故障の検出条件を以下に与える。

[補題 3.1] 任意のセルに対し '0' と '1' それぞれの書込み，読出しを行えばセル縮退故障，読出し・書込み回路縮退故障を検出できる。

[補題 3.2] 結合している2つのセル C_i , C_j 間で， C_i の任意の内容に対し， C_j のセルの内容を書換えた後， C_i の内容を読出すことで結合故障を検出できる。

[補題 3.3] 基底セルの任意の内容に対し，その隣接近傍セルの任意のパターンについて1個のセルの内容を書換えた後，基底セルのデータを読出すことで静的パターン依存故障，動的パターン依存故障を検出できる。

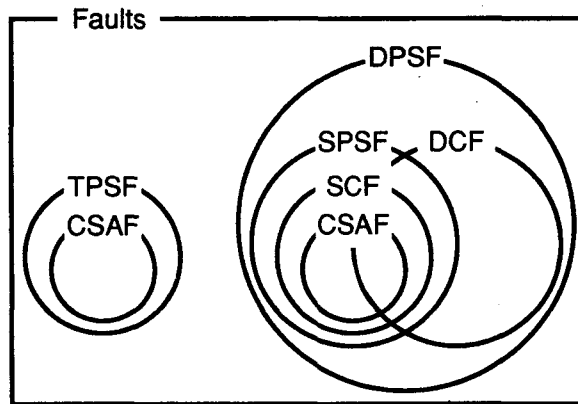
[補題 3.4] 隣接近傍セルに任意のパターンを割当てた後，基底セルの内容を書換え，再び基底セルのデータを読出すことで遷移パターン依存故障を検出できる。

[補題 3.5] '0' ('1') の読出し，'1' ('0') の書込みをすべての論理アドレスについて行えばデコーダの故障を検出できる。

[定理 3.1] 補題 3.1 ~ 3.5 は，メモリの故障を検出するための十分条件である。

(証明) 故障定義とその仮定より自明である。 (証明終)

なお，故障検出条件からみた場合，セルアレイの故障の包含関係は図 3.4 のように表すことができる。但し，結合故障については，互いに異なる近傍セルに属する任意の2個のセルが補題 3.2 を満足するものとして表している。図 3.4 より，動的パターン依存故障と遷移パターン依存故障とを検出できれば，定義したセルアレイのすべての故障を検出できることがわかる。よって，セルアレイのテストでは，動的パターン依存故障と遷移パターン依存故障とを対象としたテストについて考える。



CSAF : Cell Stuck-At Fault
 SCF : Static Coupling Fault
 DCF : Dynamic Coupling Fault
 SPSF : Static Pattern-Sensitive Fault
 DPSF : Dynamic Pattern-Sensitive Fault
 TPSF : Transition Pattern-Sensitive Fault

図 3.4 セルアレイの故障の包含関係

3.2 テスト方式

3.2.1 セルアレイ

本論文で用いるセルアレイのテストの基本概念について説明する。図 3.5 にセルアレイの構成概略図を示す。

初めにメモリの動作について概説する。メモリに行アドレスを入力すると、行デコーダにより対応する1本のワード線が選択される。このとき、このワード線上のすべてのセルはデータの読出し、書込みが可能な状態となる。次に、列アドレスを入力すると列デコーダ、セクタにより対応する1本のビット線が選択され、これと先に選択されたワード線との交点にあるセルに対して実際にデータの読出し、書込みが行われる。

以上のことから、ワード線を1本選択しておいた状態で、セルアレイとセクタとの間のすべてのビット線に対し直接データの読出し、書込みを行うことで、1本のワード線上のすべてのセルのデータを一度に扱うことができる。すなわち、1ビット/ワードのメモリにおいて、通常1回の読出し操作、書込み操作では1個のセルしかアクセスできないのに

対し、上記のようなテスト方式を用いることで1回の操作で1ワード線上のすべてのセル、例えば、正方アレイでは N^2 個のセルを同時にアクセスできるようになる。従って、テスト時間を大幅に減少させることができる。

通常、メモリではアドレス・スクランブルが行われている。すなわち、デコーダへの入力アドレスとこのアドレスによって選択されるセルのセルアレイ上での物理的位置との対応が不規則である。従って、セルアレイをデコーダと完全に分離してテストするために行デコーダとセルアレイとの間にシフトレジスタを組み込み、行デコーダのかわりにこれでワード線を選択を行う。

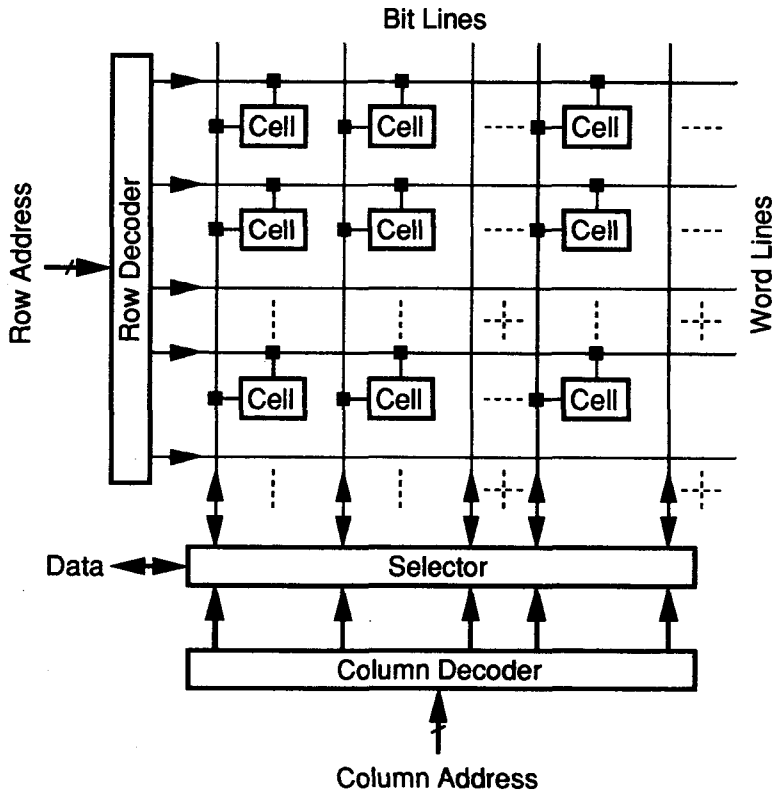


図 3.5 セルアレイの構成概略図

また、テストでは、テストベクトルが大量に必要であるため応答データも膨大な量になり、組込みテストには不利である。よって、本論文ではコンパクトテストを用い、付加ハードウェア量の減少をはかる。

なお、セルアレイのテストの説明を簡単にするため、セルアレイには図3.6に示すように行番号 m ($m = 1, 2, \dots, N^{1/2}$) と列番号 cn ($cn = 1, 2, \dots, N^{1/2}$) をつけて考える。特にことわらない限り、行(列)番号は下(右)方向に昇順になっているものとする。

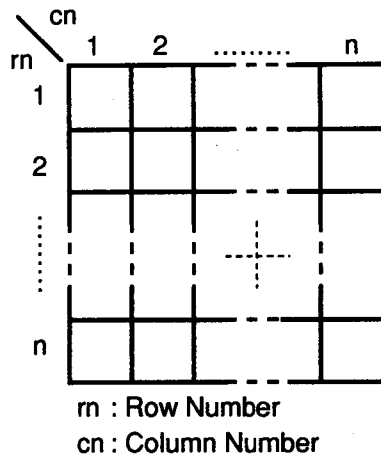


図 3.6 行番号、列番号の割当て

3.2.2 デコーダ，読出し・書込み回路

デコーダをテストするには定義3.5で示したように入力と出力との対応が全単射であるか否かを調べればよい。しかし、このままテストを実行したのではテスト用回路が大きくなる。従って、本論文ではデコーダに論理アドレスを入力し、そのとき選択したワード線上あるいはビット線上にあるセルに対しデータの読出し、書込みを行いテストする。すなわち、デコーダの入力と出力との対応を、デコーダへ入力する論理アドレスとそのとき選択されるセルとの対応に写像しテストする。行デコーダと列デコーダは個々にテストし、一方をテストするとき、これにはカウンタより順次論理アドレスを印加し、他方には一定の論理アドレスを印加する。

デコーダのテストの際、セルに対するデータの読出し、書込みには読出し・書込み回路を用い、これも同時にテストする。なお、このとき読出し・書込み回路縮退故障は無選択故障と等価な故障として検出される。

また、デコーダのテストにおいてもセルアレイのテストと同様にコンパクトテストを用い、出力応答データの判定を容易にする。

3.3 テスト系列

3.3.1 セルアレイのテスト系列

図 3.4 から、動的パターン依存故障、遷移パターン依存故障の検出条件，すなわち，補題 3.3, 3.4 を満足すればすべてのパターン依存故障を検出できる．また，隣接する 2 個のセル間の結合故障のみを対象とした場合，故障定義から明らかなように結合故障はパターン依存故障と等価な故障となる．

A. 下限値

初めに，本テスト方式を用いてパターン依存故障を検出するために十分なテスト系列中の遷移書込み操作数，読出し操作数の下限値を与える．1 個の近傍セル内のセル数が k ，1 行内にある 1 個の近傍セルの最大セル数が l ，セルアレイの行数が m の場合について考える．なお，本論文においては，パターン依存故障，および，メモリの仮定より $k = 5$ ， $l = 3$ ， $m = N^{1/2}$ である．

[補題 3.6] 1 個の近傍セルに対し，すべてのパターン依存故障を顕在化するためには $k \cdot 2^k$ のテストベクトルを印加すれば十分である．

(証明) 動的パターン依存故障と遷移パターン依存故障とですべてのパターン依存故障を表現できることから，動的パターン依存故障の場合と遷移パターン依存故障の場合とに分けて証明する．

(1) 動的パターン依存故障の場合：故障を顕在化するための遷移書込みを行えるセルは隣接近傍セルであり，その各セルは '0' から '1' と '1' から '0' との 2 通りのデータ遷移を行わなければならない．基底セルと遷移書込みを行うセル以外のセル数は $k-2$ であり，これらのセルのすべての可能なパターンの組合せの数は 2^{k-2} である．また，基底セルは '0' と '1' との 2 値を持つ．よって，この故障を顕在化するために十分なテストベクトル数は $2(k-1)2^{k-2} \cdot 2 = (k-1)2^k$ である．

(2) 遷移パターン依存故障の場合： $k-1$ 個の隣接近傍セルのすべての可能なパターンの組合せの数は 2^{k-1} である．この各パターンに対し，基底セルは '0' から '1' と '1' から '0' との 2 通りのデータ遷移を行わなければならない．よって，この故障を顕在化するために十分なテストベクトル数は $2^{k-1} \cdot 2 = 2^k$ である．

以上 (1), (2) より, 1 近傍セル当りのパターン依存故障を顕在化するためには $(k-1)2^k + 2^k = k \cdot 2^k$ のテストベクトルを印加すれば十分である. (証明終)

[補題 3.7] すべてのパターン依存故障を顕在化するためには, 1 個のセル当り 2^k 回の遷移書込み操作を行えば十分である.

(証明) セルアレイ内のどのセルも基底セルおよび隣接近傍セルになることができる. また, 各近傍セル内の任意のセルは '0' から '1' と '1' から '0' との 2 通りの遷移を行わなければならない. これに対し, 各近傍セル内の残りの $k-1$ 個のセルはすべての可能なパターンの組合せをとらなければならない. 従って, 1 個のセル当り $2 \cdot 2^{k-1} = 2^k$ 回のデータ遷移を行えば十分である. (証明終)

[定理 3.2] すべてのパターン依存故障を顕在化するために十分な遷移書込み操作数は $l \cdot 2^k \cdot m$ である.

(証明) 補題 3.7 より, 1 個のセル当り 2^k 回の遷移書込みを行えば十分である. 本テスト方式では, 1 回の書込み操作で 1 行内のすべてのセルのデータ遷移を行えるが, 故障検出条件, 補題 3.3, 3.4 より, パターン依存故障を顕在化するためには, 同一近傍セル内では 1 個のセルのみのデータ遷移を行わなければならない. このため 1 行内に同一近傍セル内の 1 個のセルが存在する場合, この近傍セルに対し, 任意のデータ遷移あるいはパターン割当てを行うためには, 1 回遷移書込み操作を繰返さなければならない. また, セルアレイの行数は m である. 従って, すべてのパターン依存故障を顕在化するためには $l \cdot 2^k \cdot m$ 回の遷移書込みを行えば十分である. (証明終)

[定理 3.3] すべてのパターン依存故障を検出するために十分な読出し操作数は $k \cdot 2^k \cdot m$ である.

(証明) 補題 3.6 より, 1 個の近傍セル当り $k \cdot 2^k$ のテストベクトルを印加すれば十分である. セルアレイ内のすべてのセルは基底セルになることができ, また, 本テスト方式では, 同一行内のすべてのセルのデータは 1 回の読出し操作で読出すことができる. 同一行内の各セルを基底セルとする近傍セルに $k \cdot 2^k$ のテストベクトルを印加し, それぞれのテストベクトルに対し基底セルのデータの読出しを行うことでこれらの近傍セルのすべてのパターン依存故障を検出できる. すなわち, 同一行に割付けられたすべての近傍セルをテストできる. また, セルアレイの行数は m である. 従って, すべてのパターン依存故障を検出するためには $k \cdot 2^k \cdot m$ 回の読出し操作を行えば十分である. (証明終)

[系 3.1] メモリに対するテスト系列が、 $l \cdot 2^k \cdot m$ 回の遷移書込み操作と $k \cdot 2^k \cdot m$ 回の読出し操作を含んでいけばすべてのパターン依存故障を検出できる。

B. テスト手続き

セルアレイ上への近傍セルの割付けを図 3.7 に示す。これは基底セル，隣接近傍セルをセルアレイ上に固定していないため，任意のセルが常に基底セル，隣接近傍セルいずれにもなることができる。このため，同一のテスト手続き内で任意のセルが基底セル，隣接近傍セルいずれにもなり得るのでテスト手続きを簡潔にすることができると考えられる。

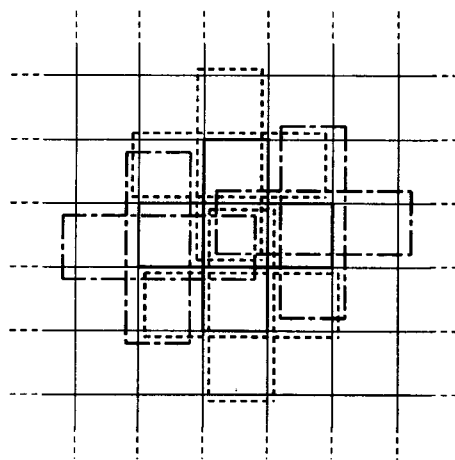


図 3.7 近傍セルの割付け

本論文で用いるテスト方式では一度に 1 ワード線上のすべてのセルのデータを扱えるが，パターン依存故障のテストにおいては同一近傍セル内の Y, B, Z とラベル付けされた隣接近傍セルのすべてには 1 回の書込み操作でデータを割当ててすることはできない。従って，W, X とラベル付けされた隣接近傍セルのデータを遷移させたときの動的パターン依存故障を検出するテスト手続きと，Y, B, Z のデータを遷移させたときの動的パターン依存故障および遷移パターン依存故障を検出するテスト手続きとに分けてパターン依存故障のテストを行う。

B-1. W, X のデータ遷移に対するテスト

テスト手続き CA-WX-T を図 3.8 に示す. この手続きでは, 初めに 1 行おきに W, X に対応する行のセルを初期化し, 次に Y, B, Z に対応する初期化しなかった行に 3 ビット・カウンタからの値を割当てて, W, X に対応する行のデータを遷移させ, W, X のデータ遷移による動的パターン依存故障の顕在化を行い, 順次テスト方向側に隣接した Y, B, Z に対応する行のデータを読み出す. このとき W のデータ遷移による動的パターン依存故障を検出できる. 再び, Y, B, Z に対応する行のデータを読み出し, X のデータ遷移による動的パターン依存故障についてテストする. 遷移するデータを補数であるものにし, 同様の操作を行う. また, テスト方向を逆にし, すなわち, W, X 間でデータ遷移を行うセルを交換し同様にテストする. 以上の操作をカウンタからのすべての出力値, すなわち, Y, B, Z のすべてのデータについて行う. ここで, 3 ビット・カウンタからのすべての値を Y, B, Z に対応する行に割当てると, この行の任意の連続した 3 列, すなわち, この行に基底セルがある任意の近傍セルの Y, B, Z には, 8 通りのパターンが割当てられたことになる. W, X と Y, B, Z との対応行を交換し同様の操作を繰り返す. なお, 出力系列のデータ圧縮を容易にするために, カウンタには 3 ビット・グレイコード・カウンタを用いる.

B-2. Y, B, Z のデータ遷移に対するテスト

同一近傍セル内の Y, B, Z すべてに 1 回の書込み操作でデータを割当てるとパターン依存故障の顕在化を行えない. 従って, 初めに同一近傍セル内にある Y, B, Z のデータ遷移について述べる.

いま, 頂点が 3 桁の 2 進数字 (頂点数 8) であり, ハミング距離が 1 である任意の 2 個の頂点 X_i, X_j において, X_i から X_j へと X_j から X_i への異なる辺が存在する有向グラフを考える (図 3.9 (a)). このグラフにおいて, 頂点は Y, B, Z のパターンに, 辺はそのデータ遷移に対応する. 各頂点において, 入次数と出次数とが等しいのでこのグラフにはオイラー経路が存在し, これに沿ってデータ遷移を行えば最適なデータ遷移書込み系列となる. このオイラー経路の長さは $3 \times 2^3 = 24$ であり, これが最適なデータ遷移書込み系列長となる. 本論文は, その経路を見つけることを目的としていないので, ここでは Hayes が用いている方法⁽²⁹⁾に従いオイラー経路を作成する. このオイラー経路 EP を図 3.9 (b) に示す.

```

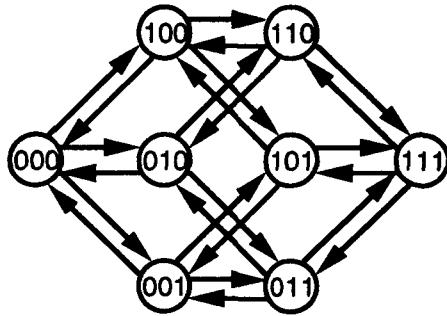
procedure CA-WX-T
  /* Test procedure for cells W and X */
  begin
    a = 0, b = 1;
    c = 1, d = N1/2;
    count = 0;
    for i = 0 to 1 do begin
      for rn = c to d do begin
        if rnmod 2 = a then
          Wm(0);
        end
      repeat
        for rn = c to d do begin
          if rnmod 2 = b then
            Wm(count);
          end
        for j = 0 to 1 do begin
          for k = 0 to 1 do begin
            for rn = c to d do begin
              if rnmod 2 = a then
                Wm;
              else
                Rm;
              end
            for rn = c to d do begin
              if rnmod 2 = b then
                Rm;
            end
          end
        end
        interchanging c and d;
      end
      count += 1;
    until count = 7
    interchanging a and b;
  end
end;

```

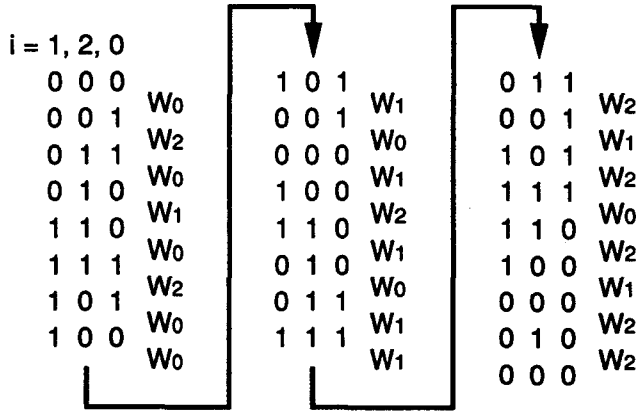
$W_a(b)$: Write data b into cells with row number a

W_c : Write complement data of content into cells
with row number c

図 3.8 テスト手続き CA-WX-T



(a) データ遷移グラフ



(b) オイラー経路 EP

図 3.9 Y, B, Z のデータ遷移グラフとそのオイラー経路

テスト手続き CA-YBZ-T とその初期化パターンとをそれぞれ図 3.10 (a), 図 3.10 (b) に示す。この手続きでは、初めに 1 行おきに Y, B, Z に対応する行のセルを初期化し、次にこれと異なる W, X に対応する行にデータ INITWX を割当てる。Y, B, Z に対応する行のデータを図 3.9 (b) の経路に沿って遷移させ、この行のデータを読み出す。このとき、データ遷移を行った列に基底セルがある近傍セルでは遷移パターン依存故障を、これ以外の近傍セルでは Y あるいは Z のデータ遷移による動的パターン依存故障を検出できる。この操作を Y, B, Z の任意のデータ遷移に対して行う。W, X に対応する行の任意のデータ遷移について同様に繰り返す。以上の操作を W, X と Y, B, Z との対応行を交換し同様に繰り返す。なお、同一行内へのデータの書込みは 2 行おきに行う。

```

procedure CA-YBZ-T
  /* Test procedure for cells Y, B and Z */
  begin
    a = 0, b = 1;
    for i = 0 to 1 do begin
      for rn = 1 to N1/2 do begin
        if rnmod 2 = a then
          Wm(0);
        end
      repeat
        for rn = 1 to N1/2 do begin
          if rnmod 2 = b then
            Wm(INITWX);
          end
        repeat
          for rn = 1 to N1/2 do begin
            if rnmod 2 = a then
              Wm,i, Rm;
            end
          until assign all patterns along the Eulerian path EP
        until assign all INITWX
        interchanging a and b;
      end
    end;
  
```

W_{a,b} : Write data into cells with row number a
and column number cn_{mod 3} = b, (b = {0,1,2})

(a) テスト手続き CA-YBZ-T

rn = 1	0	1	1	0
(2)	0	0	1	1
		0	1	1
		⋮	⋮	⋮
		⋮	⋮	⋮
rn = N ^{1/2} -1	0	0	1	1
(N ^{1/2})				

(b) 初期化パターン INITWX

図 3.10 手続き CA-YBZ-T

3.3.2 デコーダ，読出し・書込み回路に対するテスト系列

3.2.2項で述べたようにデコーダのテストは論理アドレスをデコーダに入力し，そのとき選択したセルに対しデータの読出し，書込みを行い実行する．テスト手続き D-T を図 3.11 に示す．初めに行デコーダのテスト，次に列デコーダのテストを行う．なお，セルアレイが正常な場合，デコーダのテストをセルアレイのテストに引続いて行えばすべてのセルは '0' か '1' になっているはずであり，デコーダのテストにおいて特にセルアレイを初期化する必要はない．

```

procedure D-T
  /* Test procedure for decoder and read/write logic */
  begin
    for i = 0 to 1 do begin
      column address = a;
      for ra = 0 to N1/2-1
        Rra, Wra;
      end
      for j = 0 to 1 do begin
        row address = b;
        for ca = 0 to N1/2-1
          Rca, Wca;
        end
      end
    end;

    a, b : any Logic Address (0 ≤ a, b ≤ N1/2-1)
  
```

図 3.11 テスト手続き D-T

3.4 出力系列の圧縮

3.4.1 セルアレイのデータ圧縮

[定義 3.7] 集合 S_1 , S_2 , S_0 を次のように定義する．

$$S_1 = \{cn \mid cn_{\text{mod } 3} = 1\} \quad (3.1)$$

$$S_2 = \{cn \mid cn_{\text{mod } 3} = 2\} \quad (3.2)$$

$$S_0 = \{cn \mid cn_{\text{mod } 3} = 0\} \quad (3.3)$$

ここで,

$$|S_1| = (N^{1/2} - N_{\text{mod } 3}^{1/2})/3 + i, \quad i = \begin{cases} 0 : N_{\text{mod } 3}^{1/2} = 0 \\ 1 : N_{\text{mod } 3}^{1/2} \neq 0 \end{cases} \quad (3.4)$$

$$|S_2| = (N^{1/2} - N_{\text{mod } 3}^{1/2})/3 + j, \quad j = \begin{cases} 0 : N_{\text{mod } 3}^{1/2} = 2 \\ 1 : N_{\text{mod } 3}^{1/2} \neq 2 \end{cases} \quad (3.5)$$

$$|S_0| = (N^{1/2} - N_{\text{mod } 3}^{1/2})/3 \quad (3.6)$$

である。但し、 cn は列番号を、 $|S|$ は集合 S の要素数を表す。

[定義 3.8] '0', '1' からなる幅が n , 長さが m である出力系列を R とする。

$$R = \begin{matrix} \Gamma_{11} & \Gamma_{21} & & \Gamma_{m1} \\ \Gamma_{12} & \Gamma_{22} & \cdots & \Gamma_{m2} \\ & \vdots & & \\ \Gamma_{1n} & \Gamma_{2n} & & \Gamma_{mn} \end{matrix}$$

また、各 n ビット幅のデータのビット位置を 3 を法とする剰余に分類し、それぞれを R_1, R_2, R_0 とする。 $R_1 \sim R_0$ の各 n ビットのデータの論理積、論理和否定をとった系列をそれぞれ $P_1 \sim P_0, Q_1 \sim Q_0$ とする。

$$P^1 = p_1^1 p_2^1 \cdots p_m^1 \quad Q^1 = q_1^1 q_2^1 \cdots q_m^1$$

$$P^2 = p_1^2 p_2^2 \cdots p_m^2 \quad Q^2 = q_1^2 q_2^2 \cdots q_m^2$$

$$P^0 = p_1^0 p_2^0 \cdots p_m^0 \quad Q^0 = q_1^0 q_2^0 \cdots q_m^0$$

ここで,

$$\begin{aligned}
 p_i^1 &= \bigwedge_{j=1}^{3k+1} r_{ij} & q_i^1 &= \bigwedge_{j=1}^{3k+1} \overline{r_{ij}} & r_{ij} &= \{r_{ij} \mid j_{\text{mod}3} = 1\} \\
 p_i^2 &= \bigwedge_{j=2}^{3k+2} r_{ij} & q_i^2 &= \bigwedge_{j=2}^{3k+2} \overline{r_{ij}} & r_{ij} &= \{r_{ij} \mid j_{\text{mod}3} = 2\} \\
 p_i^0 &= \bigwedge_{j=3}^{3k} r_{ij} & q_i^0 &= \bigwedge_{j=3}^{3k} \overline{r_{ij}} & r_{ij} &= \{r_{ij} \mid j_{\text{mod}3} = 0\}
 \end{aligned}$$

である。但し、 $k = (n - n_{\text{mod}3})/3$ 。

また、 $R_1 \sim R_0$ それぞれの任意の1ビットの系列を T_1, T_2, T_0 とし、このパリティをとった系列を T とする。

$$T = T_1 \oplus T_2 \oplus T_0 = t_1 t_2 \dots t_a \dots t_{2a} \dots t_{2a} t_{b+1} \dots t_{b+1} \dots t_{2b+1} \dots t_{2vb+1}$$

ここで、

$$t_i = t_i^1 \oplus t_i^2 \oplus t_i^0, \quad t_i^h \in \{r_{ij} \mid j_{\text{mod}3} = h\}.$$

系列 R, T それぞれに対し C_1, C_2 なる圧縮関数を定義する。

$$\begin{aligned}
 C_1(R) &= \sum_{i=1}^m \overline{((q_i^1 \cdot q_i^2 \cdot q_i^0) \oplus (r_i^1 \cdot r_i^2 \cdot r_i^0))} \cdot f(p_i^1 = p_i^2 = p_i^0) \\
 &+ \sum_{i=1}^m \overline{((r_i^1 \cdot q_i^2 \cdot q_i^0) \oplus (q_i^1 \cdot r_i^2 \cdot r_i^0))} \cdot f(p_i^1 \neq p_i^2 = p_i^0) \\
 &+ \sum_{i=1}^m \overline{((q_i^1 \cdot r_i^2 \cdot q_i^0) \oplus (r_i^1 \cdot q_i^2 \cdot r_i^0))} \cdot f(p_i^2 \neq p_i^1 = p_i^0) \\
 &+ \sum_{i=1}^m \overline{((q_i^1 \cdot q_i^2 \cdot r_i^0) \oplus (r_i^1 \cdot r_i^2 \cdot q_i^0))} \cdot f(p_i^0 \neq p_i^1 = p_i^2) \tag{3.7}
 \end{aligned}$$

但し、

$$p_i^h \in P_i^h, \quad (h = 1, 2, 0).$$

$$C_2(T) = \sum_{j=1}^u \left(\left(\sum_{i=2a(j-1)+1}^{2a(j-1)+a} t_i \right) \left(\sum_{i=2a(j-1)+a+1}^{2a(j-1)+2a} \bar{t}_i \right) \right) + \sum_{k=1}^v \left(\left(\sum_{i=2b(k-1)+1}^{2b(k-1)+b+1} \bar{t}_i \right) \left(\sum_{i=2b(k-1)+b+1}^{2b(k-1)+2b+1} t_i \right) \right) \quad (3.8)$$

但し、 \wedge は論理積演算記号、 $+$ 、 Σ は算術演算記号、 f は括弧内の述語が真のとき 1、偽のとき 0 の値をとる関数で、 $l = 2ua$ 、 $m = 2vb+l$ (a 、 b は任意の整数) である。

圧縮関数 C_1 は、系列 $R_1 \sim R_0$ それぞれの平行ル・データ $r^1 = \{r_{ij} \mid j_{\text{mod}3} = 1\}$ 、 $r^2 = \{r_{ij} \mid j_{\text{mod}3} = 2\}$ 、 $r^0 = \{r_{ij} \mid j_{\text{mod}3} = 0\}$ が 1 個でも異なる場合を関数 f の述語で表された 4 種類の場合に分けて計数する関数である。また、圧縮関数 C_2 は $T_1 \sim T_0$ のパリティをとり、このパリティをとった系列 T に対し、 $2a(j-1)+1 \leq i \leq 2a(j-1)+a$ 、 $2b(k-1)+b+l+1 \leq i \leq 2b(k-1)+2b+l$ のとき '1' の個数を計数し、 $2a(j-1)+a+1 \leq i \leq 2a(j-1)+2a$ 、 $2b(k-1)+l+1 \leq i \leq 2b(k-1)+b+l$ のとき '0' の個数を計数する関数である。

[定理 3.4] セルアレイが正常な場合に手続き CA-WX-T, CA-YBZ-T を用いて得られる S_1 , S_2 , S_0 それぞれの平行ル出力を圧縮関数 C_1 , C_2 で圧縮したとする。このとき C_1 , C_2 によりこれを他の出力系列と区別できる。

(証明) 正常な場合と異なる出力系列において、圧縮関数 C_1 の圧縮値が期待値と等しくなるのは S_1 , S_2 , S_0 中のすべてのデータが反転したときのみである。しかし、このときパリティは正常な場合と異なっており圧縮関数 C_2 により他の出力系列と区別できる。逆に、圧縮関数 C_2 の圧縮値が期待値と等しくなるのは 2 個のデータが反転したときであるが、このとき圧縮関数 C_1 の圧縮値は期待値と異なっており他の出力系列と区別できる。すなわち、正常な場合と異なる出力系列において、圧縮関数 C_1 , C_2 ともにその圧縮値が期待値と等しくなるような出力系列は存在しない。 (証明終)

[定理 3.5] パターン依存故障は手続き CA-WX-T, CA-YBZ-T と圧縮関数 C_1 , C_2 により検出可能である。

(証明) 手続き CA-WX-T, CA-YBZ-T はパターン依存故障を検出するための条件、補題 3.3, 3.4 を満足している。また定理 3.4 より圧縮関数 C_1 , C_2 は手続き CA-WX-T, CA-YBZ-T を用いて得られる出力系列を他の系列と区別できる。 (証明終)

[系 3.2] パターン依存故障に対し、セルアレイは手続き CA-WX-T, CA-YBZ-T と圧縮関数 C_1 , C_2 によりテスト可能であり、それぞれのテスト系列長は $105N^{1/2}$, $197N^{1/2}$ である。これらのテストの遷移書込み操作数、読出し操作数はそれぞれ $128N^{1/2}$, $160N^{1/2}$ である。出力系列を圧縮関数 C_1 , C_2 で圧縮したとき、期待値はそれぞれ 0, 0 である。

3.4.2 デコーダ，読出し・書込み回路のデータ圧縮

[定義3.9] 0, 1 からなる長さ m の系列を U とする.

$$U = u_1 u_2 \cdots u_a u_{a+1} \cdots u_{2a} \cdots t_m$$

この系列に対し，圧縮関数 C_3 を以下に定義する.

$$C_3(U) = \sum_{j=1}^u \left(\sum_{i=2a(j-1)+1}^{2a(j-1)+a} \bar{u}_i \right) \left(\sum_{i=2a(j-1)+a+1}^{2a(j-1)+2a} u_i \right) \quad (3.9)$$

但し， u_i は系列 U の i 番目のデータを表し，また， $m = 2va$ (a は任意の定数) である.

圧縮関数 C_3 は $2a(j-1)+1 \leq i \leq 2a(j-1)+a$ のとき '0' の個数を計数し， $2a(j-1)+a+1 \leq i \leq 2a(j-1)+2a$ のとき '1' の個数を計数する関数である.

[定理3.6] 手続き D-T を用いて得られる出力系列 U を圧縮関数 C_3 で圧縮したとき， U は C_3 により他の系列と区別できる.

(証明) 手続き D-T による出力系列と圧縮関数 C_3 の定義より自明である.

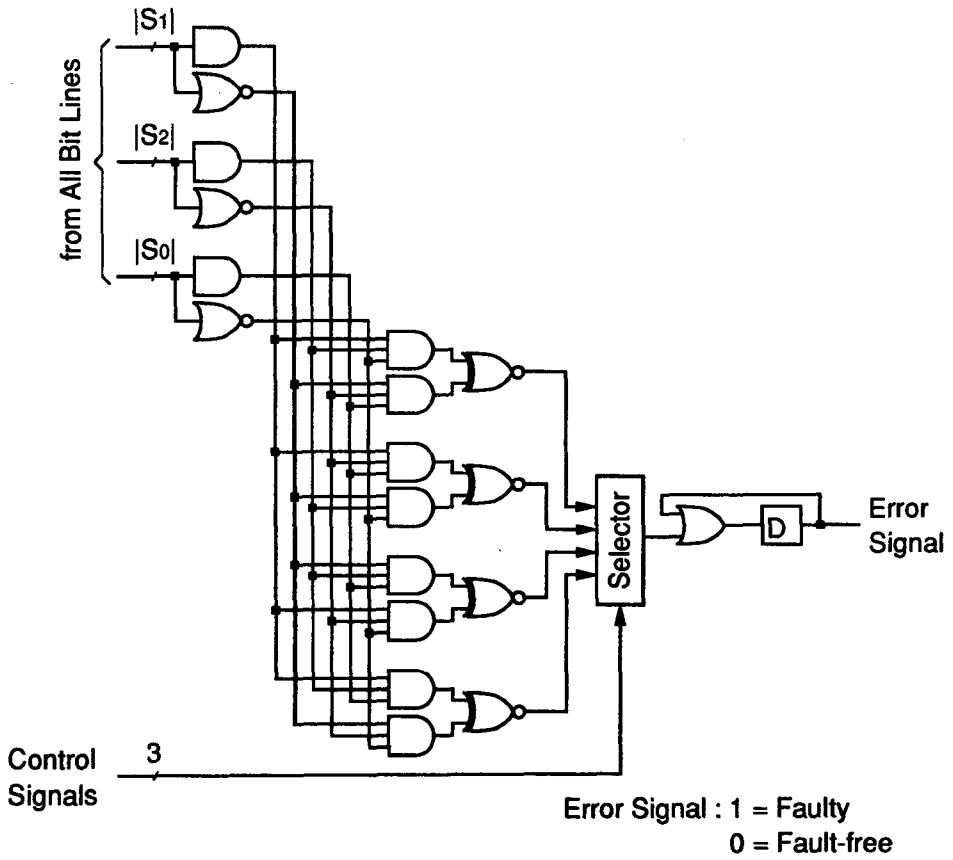
(証明終)

[定理3.7] デコーダの故障と読出し・書込み回路縮退故障とは手続き D-T と圧縮関数 C_3 により検出可能である.

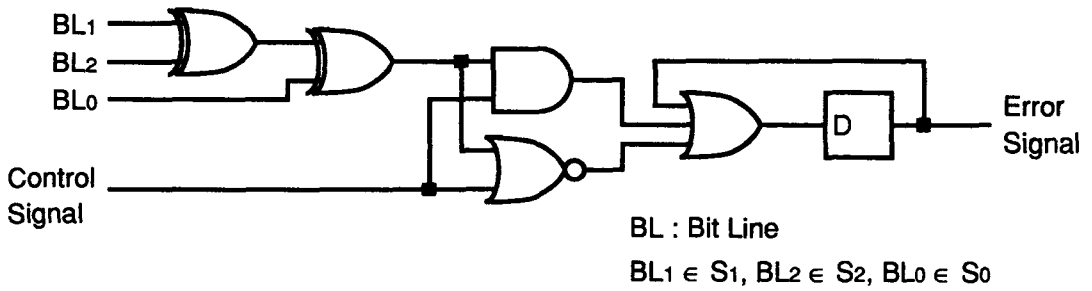
(証明) 定理3.6より，手続き D-T を用いて得られる出力系列 U は圧縮関数 C_3 により他の系列と区別できる. また，この手続きはデコーダの故障，読出し・書込み回路縮退故障を検出するための条件，補題3.1, 3.5を満足している. 従って，手続き D-T と圧縮関数 C_3 によりデコーダの故障，読出し・書込み回路縮退故障は検出可能である.

(証明終)

[系3.3] デコーダの故障，読出し・書込み回路縮退故障に対し，デコーダ，読出し・書込み回路はテスト手続き D-T と圧縮関数 C_3 によりテスト可能であり，テスト系列長は $8N^{1/2}$ である. 出力系列を圧縮関数 C_3 で圧縮したときの期待値は 0 である.



(a) 圧縮関数 C_1



(b) 圧縮関数 C_2

図 3.12 圧縮関数の実現例

3.5 テスト用回路

3.5.1 セルアレイのテスト用回路

本方式では行デコーダを使用せずにセルアレイのテストを行うため、ワード線選択の回路が必要となる。テスト手続きからわかるように、テストは奇数行、偶数行ごとに行番号の昇順、降順で行っている。これを実現するためには、行デコーダとセルアレイとの間に $N^{1/2}/2$ 段の左右シフト可能なシフトレジスタが2個必要である。但し、2個のシフトレジスタはそれぞれ奇数行、偶数行に接続する。

手続き CA-WX-T と CA-YBZ-T との書込みデータは3ビット・グレイコード・カウンタで生成できる。この他にテスト用回路の制御信号として、読出し・書込み信号、シフト方向切換え信号、書込み列制御信号、圧縮関数切換え信号などが必要である。テスト手続きは $N^{1/2}/2$ あるいは $N^{1/2}$ の系列の繰返しが基本となっており、 $\log_2(N^{1/2}/2)$ ビット2進カウンタを用いることでこれらの信号を生成できる。

圧縮関数 C_1 , C_2 をゲートで最も簡潔に実現した例をそれぞれ図 3.12 (a), 図 3.12 (b) に、セルアレイのテスト用回路のブロック図を図 3.13 に示す。

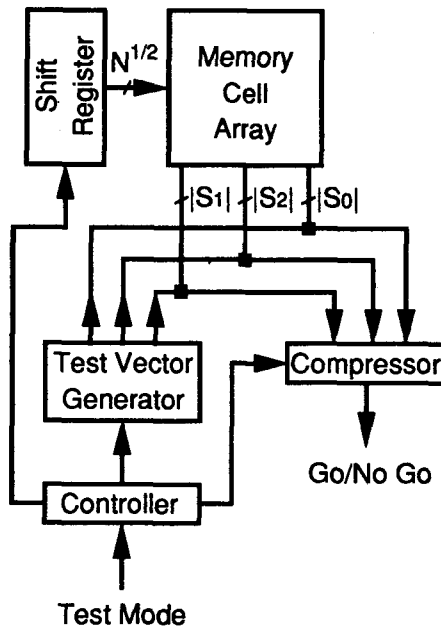


図 3.13 セルアレイのテストの回路構成

3.5.2 デコーダ，読出し・書込み回路のテスト用回路

テスト D-T は行デコーダと列デコーダとを個々にテストする。一方をテストするとき，これには論理アドレスを昇順で印加し，他方には一定の論理アドレスを印加する。正方アレイの場合，行デコーダ，列デコーダともにアドレス数は $N^{1/2}$ であるので，昇順で印加する論理アドレスは $\log_2 N^{1/2}$ ビット 2 進カウンタで生成できる。また，一定の論理アドレスとしては生成が最も容易な 0 か $N^{1/2}-1$ を用いる。このアドレスはすべてのアドレス線を電圧源に接続することで生成できる。カウンタ出力と一定論理アドレスとの切換えはマルチプレクサで行う。書込みデータは 1 個のフリップ・フロップで生成可能である。

この他にテスト用回路の制御信号として，読出し・書込み信号，マルチプレクサの切換え信号などが必要である。テスト手続き D-T は $N^{1/2}$ の系列の繰返しであるので， $\log_2 N^{1/2}$ ビット 2 進カウンタによりこれらの信号を生成できる。

圧縮関数 C_3 は図 3.14 の回路で実現できる。デコーダ，読出し・書込み回路のテスト用回路のブロック図を図 3.15 に示す。

3.6 評価

3.6.1 テスト能力

手続き CA-WX-T，CA-YBZ-T で検出可能なセルアレイの故障は以下のものである。

- (1) パターン依存故障
- (2) 縮退故障
- (3) 隣接セル間での結合故障
- (4) 奇数行に属するセルと偶数行に属するセルとの間の結合故障
- (5) 3 を法とする互いに異なる剰余類の列に属するセル間の結合故障
- (6) 任意のセル間での静的結合故障の一部

手続き D-T で検出可能な故障は以下のものである。

- (7) デコーダのすべての故障
- (8) 読出し・書込み回路の故障

また，セルアレイとデコーダ，読出し・書込み回路とのテストが独立しているためこれらの間の多重故障に対してもテスト可能である。

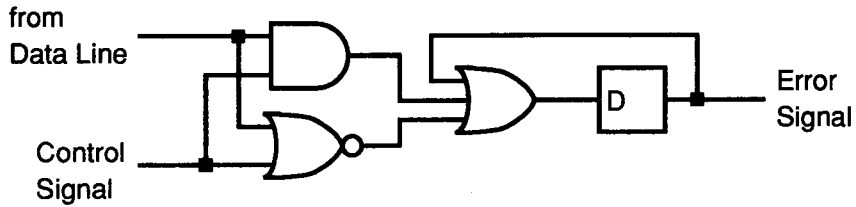


図 3.14 圧縮関数 C_3 の実現例

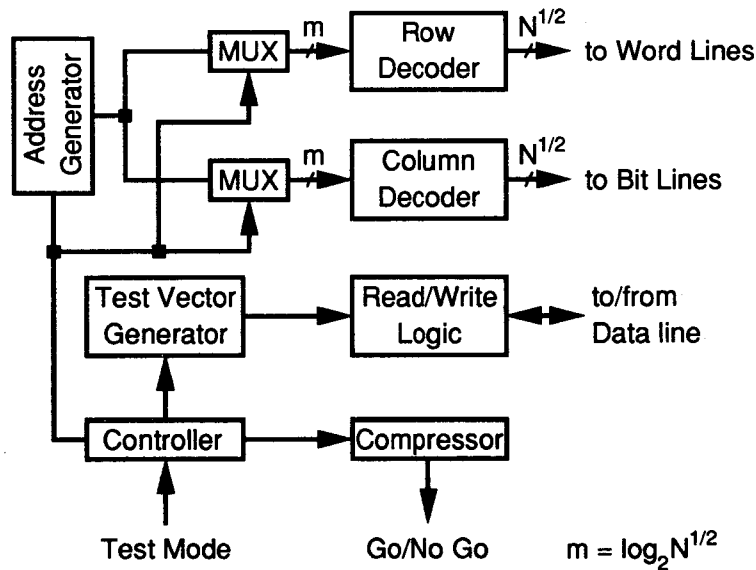


図 3.15 デコーダ、読出し・書込み回路のテストの回路構成

3.6.2 テスト時間

本論文で提案したパターン依存故障を検出するテスト手続き CA-WX-T と CA-YBZ-T によるテスト系列長は $302N^{1/2}$ である。このテスト系列では、 $128N^{1/2}$ 回の遷移書込みと $160N^{1/2}$ 回の読出しを行っている。系 3.1 と比較して $32N^{1/2}$ 回遷移書込み操作が多いが準最適なテスト系列となった。また、手続き D-T によるテスト系列長は $8N^{1/2}$ である。メモリ全体のテストは $310N^{1/2}$ のテスト系列でテスト可能であり、 $O(N^{1/2})$ のテストである。テスト時間はテスト系列長に比例するので、 $O(N)$ のテストと比較しても明らかに本方式のテスト時間が短いことがわかる。他のテスト^(3, 4, 30, 33)とのテスト系列長の比較を図 3.16 に示す。図 3.16 に示したテストのうち、本論文のテストとほぼ等しい故障検出能力を持つ Suk

と Reddy らのテスト⁽³⁰⁾と比べて、テスト時間は1Mビットで1/530、4Mビットで1/1060となり大幅に改善されている。

本論文では正方アレイを持つメモリを仮定したが、セルアレイの列数が行数より多いメモリに対してはさらに高速でテストが行える。

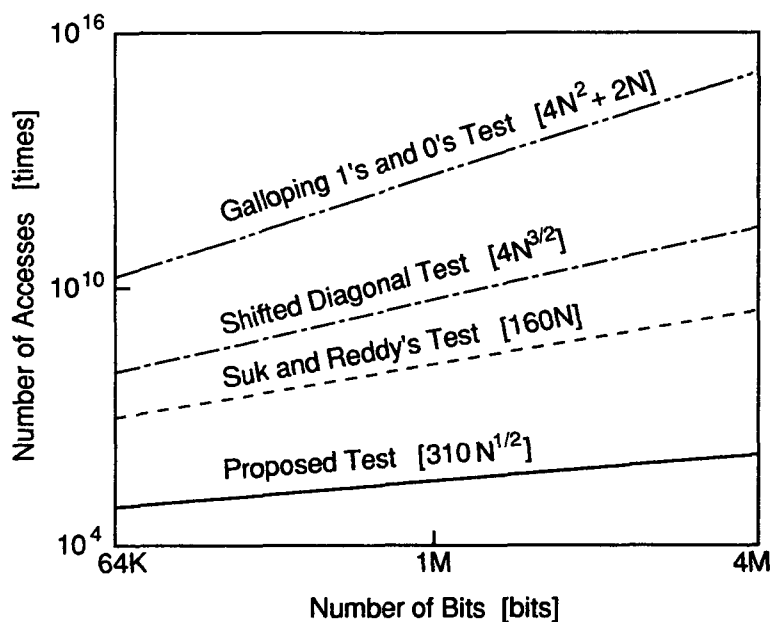


図 3.16 テスト系列長の比較

3.6.3 付加ハードウェア量

テスト用回路をゲート、フリップ・フロップで構成した場合を考える。セルアレイのテストでは $N^{1/2}/2$ 段のシフトレジスタ 2 個，デコーダ，読出し・書込み回路のテストでは $\log_2 N^{1/2}$ ビット 2 進カウンタ 1 個， $2\log_2 N^{1/2}$ 入力 $\log_2 N^{1/2}$ 出力マルチプレクサが必要である。また，圧縮回路に約 $2N^{1/2}$ ゲート，制御回路に $\log_2(N^{1/2}/2)$ ビット 2 進カウンタ 1 個およびいくつかのゲート，フリップ・フロップが必要である。ゲート 1 に対しフリップ・フロップを 4 としてゲート換算した場合，メモリの記憶容量 N に対する付加ハードウェア量の割合 ρ は，約

$$\rho = (6N^{1/2} + 14\log_2 N^{1/2} + 200)/N \quad (3.10)$$

となる。図3.17に記憶容量Nに対する付加ハードウェア量の割合を示す。記憶容量が増加するに従い付加ハードウェア量は無視できるほど小さくなることがわかる。

なお、本テスト方式を実際に1MビットDRAMに適用した結果、面積オーバーヘッドは1%以下であることが報告されている⁽⁸³⁾。

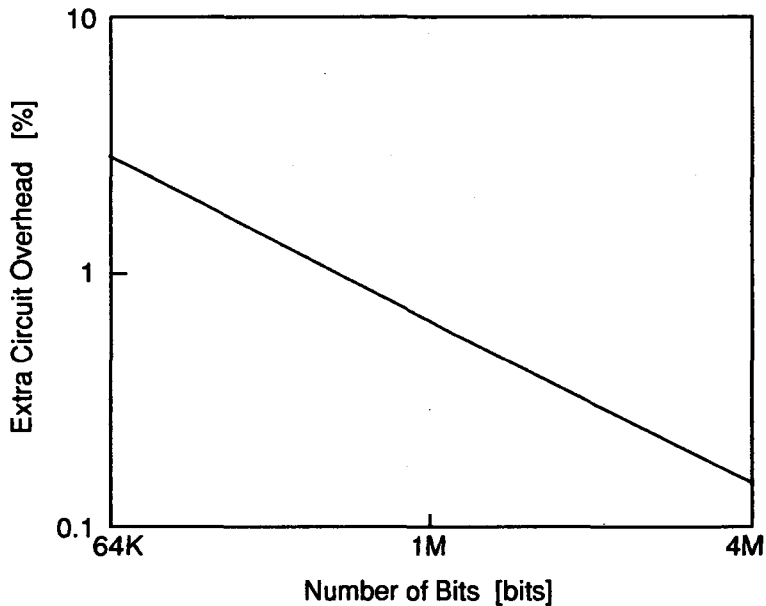


図 3.17 付加ハードウェア量

3.7 総括

本章ではメモリの組込みテストについて論じた。組込みテストでは、被テスト回路内にテスト用回路を組込むため、観測・制御点を回路内に任意に設けることができ、外部テストでは不可能な方法でテスト容易化をはかれるものと考えられる。本論文では、この考えに基づき、メモリの新しい組込みテスト方式を提案した。これは、メモリの内部動作、特に、セルアレイの動作に着目し、1本のワード線上のすべてのセルの同時テストを可能としたテスト方式である。また、この方式に適したテスト手続き、テスト用回路をも提案し

た。対象故障として、その検出が最も困難であると考えられているパターン依存故障、および、デコーダ、読出し・書込み回路の故障を考えた。これらの故障に対し、従来 $O(N)$ のテスト系列が必要であった。本論文で提案したテスト方式を用いることにより $O(N^{1/2})$ のテスト系列でテストが可能となり、大幅なテスト時間の短縮につながった。パターン依存故障に対し、提案したテスト手続きは $302N^{1/2}$ の準最適なテスト系列を生成する。さらに、本テスト方式に必要なテスト用回路は、シフトレジスタ、カウンタなどを用いて比較的簡単な機構で実現でき、しかもそのハードウェア量はメモリの記憶容量が増加するにつれて無視できるほど小さくなる。

今後は、テスト用回路の詳細な設計について検討する必要がある。また、本方式ではテスト用回路をチップ内に組込むため、その耐故障性、あるいは、自己テストをどのように実現するかについても検討を要する。

第 4 章

組合せ回路の開放故障の

テスト容易化設計

CMOS 回路固有の故障の一つにトランジスタの開放故障がある。この故障は組合せ回路を順序回路の動作にするため、その検出には連続する 2 つのテストベクトルから成るテストが必要である。開放故障に対しては、ハザードなどによる故障検出の失敗を回避するために、ローバスト性を考慮したテストを行わなければならない。本章では、ローバスト性を有する開放故障検出のためのテスト容易化設計について論じる。

4.1 節で対象故障モデルの定義とその検出条件について述べる。4.2 節で提案するテスト方式、実装方法、内部テスト点の選択方法について述べた後、4.3 節ではテスト生成アルゴリズムとベンチマーク回路に対するテスト生成の実験結果とを示す。4.4 節で本章のまとめを行う。

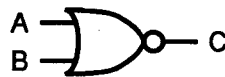
本論文で対象とする組合せ回路は、完全相補形 CMOS 基本ゲート、すなわち、NAND ゲート、NOR ゲート、NOT ゲートのみからなる CMOS 組合せ回路であると仮定する。

4.1 故障モデルと検出条件

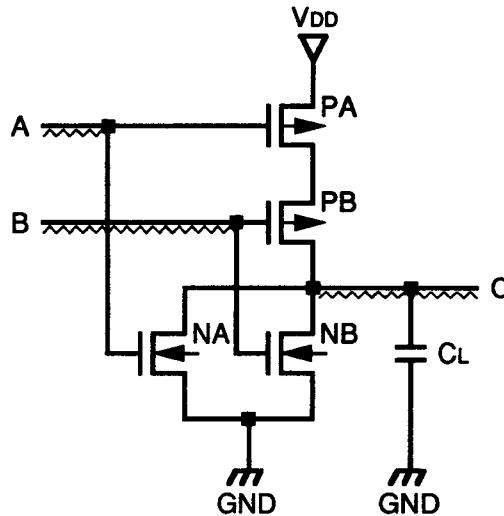
本節では、対象とするゲートレベルとトランジスタ・レベルの故障定義とその検出条件について述べる。なお、故障はメモリと同様に単一固定故障のみを考える。

初めに故障モデルを定義する。

[定義 4.1] ゲートの入出力信号線の論理値が、常に論理値 '0' ('1') に固定される故障を信号線の 0 (1) 縮退故障 (Stuck-At-0 (1)) と呼ぶ。なお、0 縮退故障と 1 縮退故障とを特に区別しない場合は、単に縮退故障 (Stuck-At Fault) と呼ぶ。



(a) ゲート記述



(b) トランジスタ記述

図 4.1 2 入力 CMOS NOR ゲート

図 4.1 (a) のゲート記述の 2 入力 CMOS NOR ゲートにおいて、入出力信号線 A, B, C は、トランジスタ記述の回路では図 4.1 (b) の波線の部分に対応する。

[定義 4.2] MOS トランジスタのゲート端子の電圧値にかかわらず、常にトランジスタが開放状態となる故障を開放故障 (Stuck-Open Fault) と呼ぶ。

次に定義 4.1, 4.2 の故障を検出するための条件を以下に示す。

[補題 4.1] ゲート出力に、正常時と故障時とで異なる論理値を出力させる入力をゲートに印加したならば、ゲートの入出力線の縮退故障を顕在化できる。

[補題 4.2] ゲート出力をハイ・インピーダンス状態にする入力をゲートに印加したならば、ゲート内の開放故障を顕在化できる。

[定理 4.1] 開放故障を顕在化する入力を印加する直前に、その入力による正常な場合の出力値と補数となる出力を与える入力を印加したならば開放故障を検出できる。

(証明) 検出ベクトルを印加したとき、開放故障が存在するゲート出力は直前に印加されたテストベクトルによる論理値を保持する。従って、検出ベクトルを印加する直前に、初期化ベクトルを印加すれば開放故障を検出できる。 (証明終)

定理 4.1 より、開放故障を検出するためには、通常、連続する 2 つのテストベクトルから成るテストが必要であることがわかる。

4.2 開放故障のテスト法

4.2.1 ローバストテスト手法

2.5 節では 2 パターンテストであってもローバストテストを用いることにより、タイミング・スキューや回路遅延の影響を受けずに開放故障のテストを正常に行えると述べたが、ローバスト性を有するテストベクトルの生成は困難であり、また、回路構成によっては必ずしもローバスト・テストベクトルを生成できるわけではない。

2 パターンテストでは、負荷容量の充放電電圧を利用して開放故障の検出を行っているが、ここにローバスト性に関し問題があると考えられる。すなわち、2 パターンテストではゲート出力がハイ・インピーダンス状態になったとき、出力が初期値を保持しているか否かという動作を利用している。よって回路動作の観点からとらえた場合、この回路は不安定な状態にあり、回路遅延などの周囲の影響を受け易いという問題がある。

開放故障が顕在化され出力がハイ・インピーダンスになったとき、このゲートは見かけ上、トライ・ステート素子となっている。このとき、このゲート出力は外部から任意の論理値に設定可能となる。従って、あるテストベクトルを印加したとき、ゲート出力がハイ・インピーダンス状態であるかを確認できればローバスト性を有する開放故障のテストが可能になるものと考えられる。

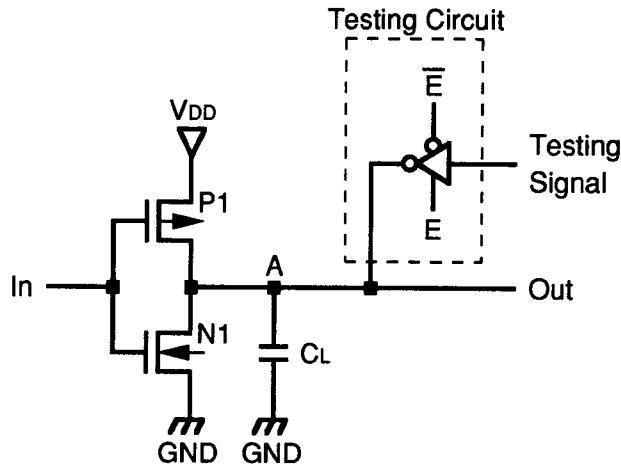


図 4.2 テスト用回路

まず、本論文で提案するローバストテスト方式について説明する。提案する方式では、被テストゲートにテストベクトルを印加しておいた状態で、回路外からこのゲート出力を '0'、および、'1' に直接駆動し、その結果、出力論理値がどうなるかを観測する。図 4.2 に本方式のテスト用回路を示す。図 4.2 では説明を簡単にするため被テストゲートをインバータとした。ゲート出力にはテスト用信号の印加ドライバであるトライ・ステート・インバータをテスト用に付加してある。通常動作では $E = '0'$ 、テスト時には $E = '1'$ である。ゲート入力 In にテストベクトルを印加しておいた状態で、テスト信号として '0'、および、'1' を印加する。ここで、被テスト回路の出力への印加値として、ゲート出力の期待値に対し、補数となる論理値 '0'、または、'1' のどちらか一方のみを印加すれば良いが、各テストベクトルに対しテスト信号を制御する複雑さを回避するため '0' と '1' の両方を印加することにする。 $E = '1'$ としたとき、もしこのゲートに開放故障が存在し、かつ、出力 Out がハイ・インピーダンス状態ならば出力は '0'、および、'1' となる。しかし、ゲートが正常な場合、 $E = '1'$ としたとき節点 A で信号衝突が起き出力は不定となる可能性がある。この影響を避けるためには、テスト信号印加用ドライバの駆動能力は弱くなくてはならない。すなわち、ラッチ回路などのフィードバック部のインバータに用いられるようなトランジスタのチャンネル幅/チャンネル長の小さい駆動能力の弱いインバータをドライバとして使用する。こうすることで $E = '1'$ としたとき被テスト回路が正常な場合には、出力を被テスト回路の論理で決定される論理値に設定することができる。表 4.1 に図 4.2 に示したテスト用回路

による出力値を示す。正常時にはテスト信号にかかわらず、ゲートの入力値によって決まる論理値が出力される。開放故障が顕在化されたとき、出力は印加したテスト信号値の反転値となる。また、縮退故障が存在するとき、ゲート入力値、テスト用信号にかかわらず、'0'、または、'1'のどちらか一方の論理値が出力される。以上述べたように、本テスト方式では、開放故障の顕在化時の負荷容量による充放電電圧を利用せずに開放故障の検出を行うため、1つのテストベクトルでテスト可能であり、また、回路遅延などによるテストの無効がなくローバスタ性を有したテストが可能となる。

表 4.1 図 4.2 のテスト用回路による出力値

in	1		0	
Testing Signal	0	1	0	1
Fault-free	0	0	1	1
N1 : SOPF	1	0	1	1
P1 : SOPF	0	0	1	0
Out : SA-0 In : SA-1	0	0	0	0
Out : SA-1 In : SA-0	1	1	1	1

} Output Values

4.2.2 実装方法

被テスト回路に図 4.2 の本論文で提案したテスト用回路を適用した場合、可制御性・可観測性は高くなるが、1 ゲート当りの付加ハードウェアとしてトライ・ステート・インバータのトランジスタ4個、トライ・ステート・インバータの制御信号線2本とテスト用の信号線1本が必要となり面積オーバーヘッドが大きくなると予想される。

チップ内の任意のゲート出力を電氣的に観測可能とする技術としてクロスチェック・テスト技術が提案されている。これは図 4.3 に示すように、ゲート出力にスイッチ用 MOS ト

ランジスタを1個付加し、このトランジスタを制御してゲート出力を観測可能にするテスト技術である。スイッチ用トランジスタはクロスポイント・スイッチCPSと呼ばれ、そのゲート端子は制御信号線であるプローブ線に接続されており、また、ソース、ドレインの一方はゲート出力に、他方はゲート出力の読出し信号線であるセンス線に接続されている。

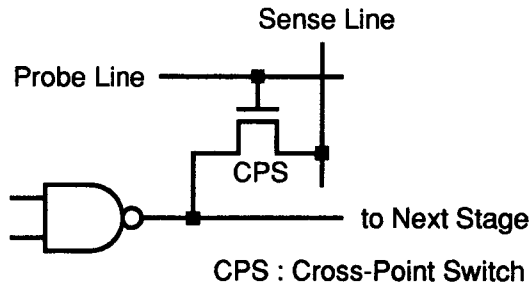


図 4.3 クロスチェック・テスト技術

少ないハードウェア量で本テスト方式を実現するために、クロスポイント・スイッチをゲート出力の制御・観測用に導入することを試みる。本テスト方式をクロスポイント・スイッチを用いて実現した場合、図 4.2 のドライバはトライ・ステート・インバータの代わりに通常の駆動能力の弱いインバータで構成でき、また、各ゲート出力にドライバを付加する必要はなく、駆動能力を十分考慮すれば同一センス線上には1個のドライバを付加すれば十分である。

クロスチェック・テスト技術の本テスト方式の回路に適用した場合のテスト手順 HOCT-P、および、チップ全体の回路構成をそれぞれ図 4.4、および、図 4.5 に示す。テスト手順は、外部入力信号線にテストベクトルを印加しておいた状態で、すべてのセンス線にテスト信号 '0' と '1' を印加する。次に、クロスポイント・スイッチをオンにする。このとき故障があればセンス線上の論理値は表 4.1 に示した値になり、故障がなければ被テストゲートの論理で決定される論理値になる。テスト用信号印加ドライバをオフさせたあとクロスポイント・スイッチをオフにし、すべてのセンス線の論理値を観測する。以上の操作をすべてのテストベクトルについて繰り返す。

なお、図 4.5 において、内部テスト点数は $m \cdot n$ である。ここで m はプローブ線数、 n は

```

procedure HOCT-P
  /* Test procedure for the proposed method */
  begin
     $v = \{0, 1\}$ ;
    while given faults exist do begin
      apply a test vector to primary inputs;
      for each  $v$  do begin
        apply testing signal  $v$  to all sense lines from driver;
        cross-point switches on;
        driver off;
        cross-point switches off;
        observe all sense lines;
      end
    end
  end;
  
```

図 4.4 テスト手順 HOCT-P

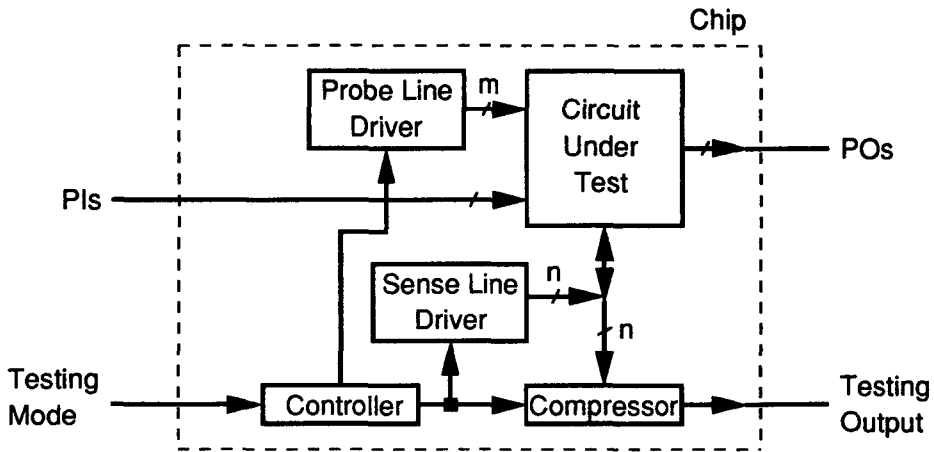


図 4.5 提案したテスト方式のブロック図

センス線数である。外部入力信号線への印加テストベクトル数を k とした場合、出力される応答系列の長さは $2 \cdot k \cdot m$ であり、また、各出力応答のビット幅は n ビットである。本方式をそのまま IC チップに適用するとピン数がかなり増加するので、チップ内で出力データを圧縮する必要がある。図 4.5 では、出力データを圧縮器で圧縮するコンパクトテストとして実現しているが、これは多入力 LFSR (Multiple-Input Signature Register : MISR) でも実現できる。図 4.6 に 4 ビット MISR の回路を示す。この回路では 4 個のレジスタに 4 本の入力信号線 I_0, I_1, I_2, I_3 から ExOR ゲートを通して並列にデータが入力する。

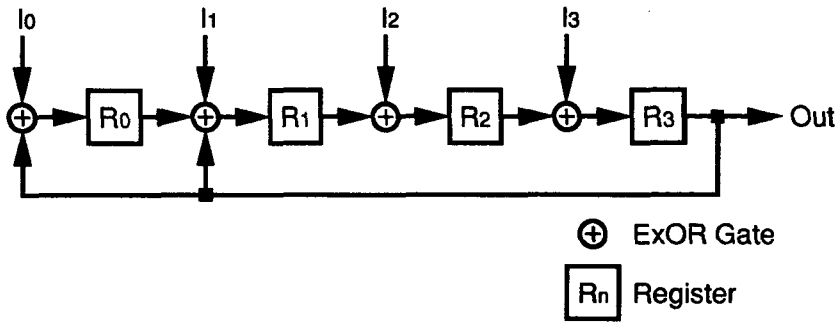


図 4.6 4 ビット MISR

単一入力 LFSR と同様に，MISR でデータ圧縮をした場合，故障の見逃しをする場合がある．また，後述する面積オーバーヘッドの算出のために， m ， n の選択と MISR に必要なレジスタのビット数との関係について考察する．MISR のレジスタのビット数を p ，MISR への入力系列長を q としたとき，単一故障のエイリアス誤り率を 0 にするためには次式を満足しなければならない．

$$p \geq \log_2 q \tag{4.1}$$

$q = 2 \cdot k \cdot m$ を代入すると次式を得る．

$$p \geq \log_2(2 \cdot k \cdot m)$$

$$2^p \geq 2 \cdot k \cdot m \tag{4.2}$$

内部テスト点数を t としたとき

$$t = m \cdot n. \tag{4.3}$$

(4.3) 式を (4.2) 式に代入すると

$$n \cdot 2^p \geq 2 \cdot k \cdot t \tag{4.4}$$

となる．

また出力は n ビット幅であるので

$$p \geq n \tag{4.5}$$

でなければならない。

$p = n$ を (4.4) 式に代入すると

$$n \cdot 2^n \geq 2 \cdot k \cdot t. \quad (4.6)$$

以上より、(4.6) 式を満足する m , n を選び、かつ、(4.5) 式から、 n ビット以上の MISR を使用しなければならない。

4.2.3 テスト点の選択

本提案のテスト方式を用いて全ゲート出力を観測した場合、1つのテストベクトルで開放故障のテストができローバスト性を有したテストが可能である。しかし、すべてのゲート出力をテスト点にすると、テスト点数が多く多層配線技術を用いたとしても付加配線量が多いためレイアウト設計が困難になるものと予想される。よって、テスト点を適当なゲート出力に設定し、これらの代表テスト点でテストすることが必要である。

回路に連続した異なる2つのテストベクトルを印加したとき、タイミング・スキューや回路遅延が存在していても必ずしもハザードが発生するものとは限らない。しかし、これを検証するためには実際の被テスト回路を忠実にモデル化した正確な回路シミュレーションを実施しなければならずかなり困難である。よって、本論文では以下の仮定が成立するものとする。

[仮定 4.1] わずかなタイミング・スキューや回路遅延によってでもハザードは発生する。

また、次に示すテストベクトルをローバスト・テストベクトルとする。

[定義 4.3] 外部入力信号線数 m の回路内の n 入力ゲート G のテストについて考える。ここで、外部入力信号線を $p_j (j = 1, 2, \dots, m)$, G の入力信号線を $i_h (h = 1, 2, \dots, n)$ とする。 $\langle T_1, T_2 \rangle$ を G 内の入力が $i_k (1 \leq k \leq n)$ であるトランジスタ Q の開放故障に対する2つのテストベクトルであるとする。外部入力信号線にテストベクトル T を印加したときの信号線 x の論理値を $T(x)$ とする。 $T_1(i_k) = v$, $T_2(i_k) = \bar{v}$, $v = \{0, 1\}$ であり、かつ、 $T_1(i_g) = T_2(i_g)$, ($i_g | 1 \leq g \leq n, g \neq k$) で、さらに、これら $T_1(i_g)$, $T_2(i_g)$ を制御している $\{p_f | 1 \leq f \leq m\}$ ($\subset \{p_j\}$) から i_g までの最低1本の経路 r 上の各信号線では $T_1(s) = T_2(s)$ である $\langle T_1, T_2 \rangle$ をローバスト・テストベクトルと呼ぶ。ここで、 s は経路 r 上の任意の信号線を表す。

定義 4.3 はハザードが発生しないための十分条件である。

[定義 4.4] 同一分岐点を起点とする複数の分岐枝経路を再び 1 本の経路にしているゲートを再収斂ゲートと呼ぶ。また、これ以外のゲートを非再収斂ゲートと呼ぶ。さらに、この分岐した経路が再び 1 本に交わる経路を再収斂経路と呼ぶ。

図 4.7 に再収斂ゲートの検索のためのアルゴリズムを示す。分岐点からの各分岐枝経路に対し、その経路上の各信号線に同一色を塗り、ゲートの入力に 2 本以上の同一色があるとき、このゲートは再収斂ゲートとなる。このプログラムは C 言語を用い、SUN4 ワークステーション上に実現した。

```

procedure SEAREG
  /* Searching the reconvergent gate */
  begin
    while unchecked fan-out points exist do begin
      for each path from fan-out point do begin
        sl = fan-out branch;
        while sl ≠ primary output do begin
          assign a color to sl;
          if gate input has had the color;
            then reconvergent gate
          sl = forward line;
        end
      end
      clear the color;
    end
  end;

```

図 4.7 再収斂ゲート検索アルゴリズム

[例 4.1] 図 4.8 に示した回路において、 RG_x ($x = 1, 2, 3$) は再収斂ゲートである。また、 NRC_y ($y = 1, 2, \dots, 5$) は非再収斂ゲートからなる非再収斂回路である。

回路内の適当なゲート出力点で信号値を観測する場合、故障による影響を伝搬させるためにローバスタな 2 パターンテストを行わなければならない。開放故障のテスト生成において、それがローバスタ性を有するようにするには一般に複雑かつ困難であるが、回路構成とテスト点をチップ内に任意に設定できるということを考慮した場合、必ずしもローバスタ・テストベクトルの生成が困難であるというわけではない。分岐・再収斂のない樹状回路を例にとると、任意のゲートの各入力を任意の論理値に制御するとき、このゲートの

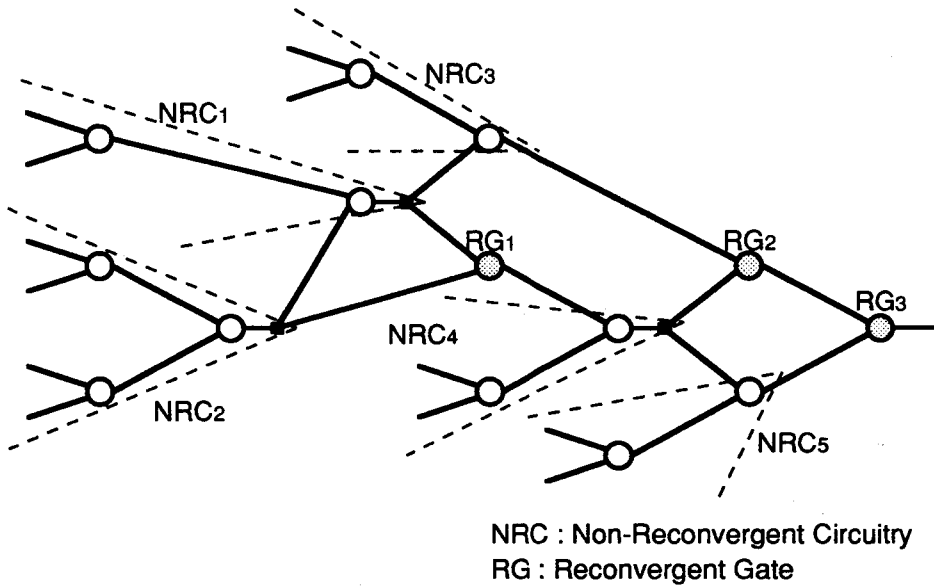


図 4.8 再収斂ゲートの回路例

他の入力に影響を与えることなく外部入力信号線への印加値を決定することができる。よって、ローバスト・テストベクトルの生成は比較的容易である。

[定理 4.2] 再収斂ゲートの出力をテスト点とすれば本テスト方式によりローバストテストを実現できる。

(証明) 非再収斂ゲートならば、その各入力の論理値はそれぞれ異なる外部入力信号線で制御できる。よって、ローバスト・テストベクトルが存在するならば、定義 4.3 に示したローバスト・テストベクトルは容易に生成可能である。一方、再収斂ゲートでは、その入力信号線のうち少なくとも 2 本は最低 1 本の同一の外部入力信号線に接続している。従って、このゲートに対して 2 パターンテストを実施したとき定義 4.3 に示したローバスト・テストベクトルを生成できない場合がある。すなわち、再収斂ゲートのある 1 本の入力信号線の論理値を遷移させる $\langle T_1, T_2 \rangle$ を外部入力信号線に印加したとき、このゲートの他の入力信号線にハザードが発生する可能性がある。この場合、再収斂ゲートの出力を本テスト方式でテストすればローバスト性のあるテストが可能となる。よって、再収斂ゲートの出力をテスト点とすれば、被テスト回路に対してローバストテストが可能である。

(証明終)

以上のことから回路構造を考慮した場合、再収斂ゲートにはローバスト・テストベクトルが存在しない場合もあり、内部テスト点として再収斂ゲートの出力を選択することがローバストテストを可能にする一方法である。

再収斂ゲートをテスト点として、本方式を加算器に対して実装した場合のラフレイアウト設計による面積オーバーヘッドを図4.9に示す。16ビット加算器では面積オーバーヘッドは約17%、256ビット加算器でのそれは約2%であり、回路規模が大きくなるにつれて面積オーバーヘッドが小さくなることがわかる。

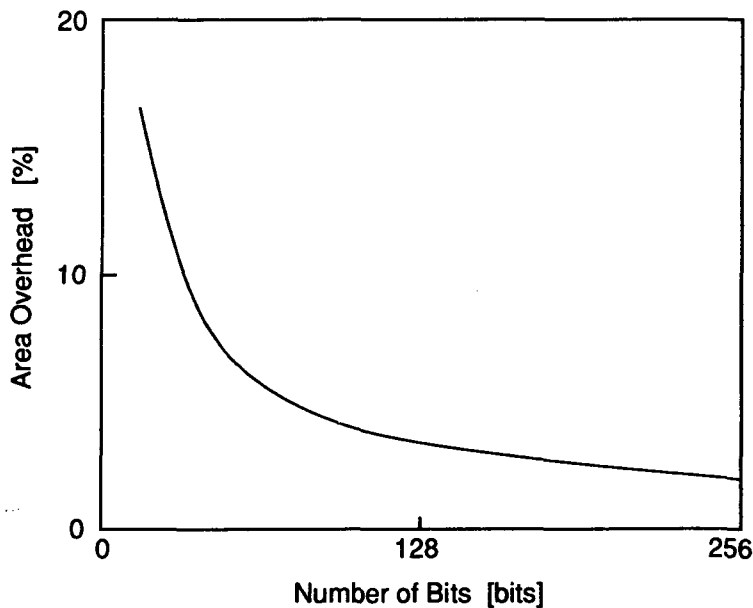


図 4.9 加算器に対する面積オーバーヘッド

4.3 テスト生成

4.3.1 テスト生成アルゴリズム

テスト生成法と実験結果について述べる。図4.10にテスト生成アルゴリズムを示す。このプログラムはC言語を用い、SUN4ワークステーション上に実現した。テスト生成の前処理として、各種ゲートを対応するトランジスタ・レベルの回路に変換し、次に、変換さ

```

procedure TPG-HOCT
  /* Test generation for HOCT */
  begin
    pre-processing;
    while given faults exist into non-reconvergent gates do begin
      generate test vector;
      if generation of test vector is successful then begin
        generate initializing vector;
        if generation of initializing vector is successful then begin
          check hazard;
          provoke fault simulation;
        end
      end
    end
    while given faults exist into reconvergent gates do begin
      generate test vector;
      if generation of test vector is successful
        then provoke fault simulation;
    end
  end;

```

図 4.10 テスト生成アルゴリズム

れた回路内の開放故障を検出するために必要な2つの入力値を求める。これらの入力値に基づいて被テスト回路に対する2テストベクトル $\langle T_1, T_2 \rangle$ を生成する。よって、対象故障はトランジスタ・レベルであるが、ゲートレベルでのテスト生成が可能である。

以上の前処理を終了したのち、非再収斂ゲート内の故障に対しては、再収斂ゲート、または、外部出力信号線まで伝搬させるローバストな2テストベクトルの生成を、再収斂ゲート内の故障に対しては、検出ベクトルのみの生成をそれぞれ行う。2パターンテストでは、初期化ベクトルは被テストゲートの出力のみの初期化を行えば良いのに対し、検出ベクトルは故障を顕在化し、かつ、故障の影響を観測点である外部出力端子、または、内部テスト点まで伝搬しなければならない。本論文の2テストベクトル生成法では、初めに検出ベクトルを生成し、次にこれに見合う初期化ベクトルを生成する。直観的に、この方法の方がローバスト・テストベクトルを容易に生成できるものと考えて用いている。テスト生成時には、ハザード発生のチェックを行う。図 4.8 の回路において、非再収斂回路 NRC_y 内の各故障に対しては、2パターンテストを行う。故障による影響が外部出力端子、または、最低1個の再収斂ゲート RG_x の出力に伝搬すれば、故障は検出可能である。 RG_x

のテストに対しては、検出ベクトルのみを生成する。2パターンテストを生成した後、再収斂ゲートに対する検出ベクトルの生成をしているので、故障シミュレーションによって既に検出されている再収斂ゲート内の故障に対してはテスト生成を行わない。なお、各テストベクトルは、PODEM アルゴリズムに基づいたアルゴリズムを用いて生成している。

4.3.2 実験結果

表 4.2 に加算器 adder と ISCAS '85 の組合せ回路のベンチマーク回路について、非再収斂ゲートの開放故障を再収斂ゲート出力まで伝搬させた場合のローバスト・テストベクトルの生成結果を示す。ISCAS '85 のベンチマーク回路とは、1985 年の回路システム国際シンポジウム (International Symposium on Circuits and Systems : ISCAS) で発表されたテスト生成のための標準回路である。ベンチマーク回路はゲートレベルで回路が記述されている。このため AND (OR) ゲートは、NAND (NOR) ゲートの次段に NOT ゲートが接続された 2 段回路で構成されているものと仮定している。ゲートの欄は、この様な考え方に基づいた回路内のゲート数を表している。ゲート内で直列に接続されたトランジスタ、例えば図 4.1 の PA と PB との故障は、同一のテストベクトルで検出可能であるのでこれらの故障を等価故障として扱う。AND (OR) ゲートを NAND (NOR) ゲートと NOT ゲートとで構成した場合、この NOT ゲート内の故障は、その前段の NAND (NOR) ゲートに対するテストによって検出できるので、これらの NOT ゲート内の故障は NAND (NOR) ゲート内のそれと

表 4.2 テスト生成の実験結果

回路名	ゲート	再収斂ゲート	代表故障	テスト不能故障*	ベクトル	検出率 [%]	ハザード
adder	76	10	121	0 (0)	126	100.0	free
cm432	164	92	478	1 (1)	328	99.8	free
cm449	260	82	506	0 (0)	454	100.0	free
cm880	555	133	1112	0 (0)	1304	100.0	free
cm1355	636	384	1610	8 (0)	855	99.5	free
cm1908	1097	250	2370	7 (0)	1404	99.7	free
cm3540	2482	600	4608	170 (4)	3078	96.3	free

*: () 内はバックトラック 10 回でテスト生成を打切った故障数であり、内数である。

等価であると考え、以上の考え方に基づいた代表故障数を代表故障の欄に示している。テスト不能故障数はテストベクトルの存在しない故障数を表している。ここで、()内の数は、バックトラック 10 回でテスト生成を打切った故障数であり、内数である。ベクトルの欄は生成されたテストベクトル数を表している。故障検出率 ρ は次式により算出した。

$$\rho = (1 - (\text{テスト不能故障数} / \text{代表故障数})) \times 100 \quad (4.7)$$

実験結果より、次のことが明らかになった。すべてのテストはハザードが発生していないのでローバストテストとなっており、また、故障検出率も高い。さらに、再収斂ゲートの出力をテスト点とすることで、テスト点数を全ゲート数の 13 ~ 60 % に減少させることができた。

4.4 総括

本章では、CMOS 組合せ論理回路のトランジスタの開放故障に対するテスト容易化設計の提案とテスト生成について論じた。本論文では、開放故障が顕在化されたとき、ゲートは見かけ上トライ・ステート素子になることを利用し、開放故障を顕在化させるテストベクトルを印加したあとに回路外部からゲート出力をアクセスするテスト方式の提案を行った。この方式はハザードなどの影響を受けずに開放故障のテストを行えるローバストテストである。本方式は縮退故障にも適用できる。さらに、回路内部の再収斂ゲート出力点をテスト点とし、テスト点数の減少をはかった。これによりテストのローバスト性を失うことなくテスト点数を全ゲート数の 13 ~ 60 % に減少させることができ、付加ハードウェア量の減少をはかれた。加算器に対し本方式のレイアウト設計を行なった結果、回路規模が大きくなるにつれて面積オーバーヘッドが小さくなることが明らかになった。テスト生成の実験結果から、生成されたテストはハザードのないローバストテストであることを確認できた。また、本テスト方式を用いることにより、従来の 2 パターンテストのみによるローバストテストに比べ、故障検出率が大幅に向上した。

今後は組込むテスト用回路の自己テスト方法やテスト点数を一層減少させるための代表テスト点の決定方法、テストベクトル数の削減などについて検討を要する。

第 5 章

組合せ回路の組込み電流テスト

CMOS 回路には，従来の回路出力の論理値のみによるテストでは検出できない故障が存在する．このうちトランジスタの導通故障やゲート酸化膜短絡故障などは，顕在化されたとき異常な静的電流を生じるため，電源電流の計測を行うことで検出できる．電流テストは CMOS 回路の故障検出に非常に有効であるが，その一方でテスト速度が遅く，また，計測電流値の分解能が低い．この問題を解決するために，組込み電流テストが提案されている．本章では，完全相補形 CMOS 組合せ回路の組込み電流テストのためのテスト用回路の設計とテスト生成法とについて論じる．

5.1 節では対象とする故障モデルと検出条件について述べる．次に，5.2 節でテスト用回路の設計について述べる．提案する回路に適したテスト系列の生成手法，および，テスト生成の実験結果について 5.3 節で述べる，最後に，5.4 節で本章のまとめを行う．

なお，本論文で述べる電流テストは，各故障に対しテストベクトルを印加し，逐次電源電流値を計測するという点で DC テストとは本質的に異なる．

5.1 故障モデルと検出条件

本節ではCMOS 組合せ回路のトランジスタ・レベルの故障定義とその検出条件について述べる。なお、故障は単一固定故障のみを考える。

初めに故障モデルを定義する。

[定義 5.1] MOS トランジスタのゲート端子の電圧値にかかわらず、常にトランジスタが導通状態となる故障を導通故障 (Stuck-On Fault) と呼ぶ。

[定義 5.2] MOS トランジスタのゲート電極とシリコン基板との短絡をゲート酸化膜短絡故障 (Gate Oxide Short Fault) と呼ぶ。

縮退故障は定義 4.1 と同様である。定義 4.1 でモデル化した縮退故障や上記した故障は、顕在化されたときに顕著な静的電流が流れるため、電流テストにより故障検出を行うことが可能である。

次に、縮退故障、導通故障、ゲート酸化膜短絡故障を電流テストにより検出するための条件を以下に示す。

[補題 5.1] 各信号線に論理値 '0' と '1' とを割当てることにより、電流テストによりすべての縮退故障とゲート酸化膜短絡故障とを検出することができる。

(証明) 故障モデルより 1 (0) 縮退故障ならばその信号線の電圧は V_{DD} (GND) であり、nMOS (pMOS) のゲート端子と基板との間のゲート酸化膜短絡故障ならばそのトランジスタのゲート端子電圧は GND (V_{DD}) である。よって、これらの信号線に対し、相補な電圧、すなわち、補数となる論理値を印加することにより、 V_{DD} - GND 間に導通経路が形成され、電流テストでテストが可能となる。 (証明終)

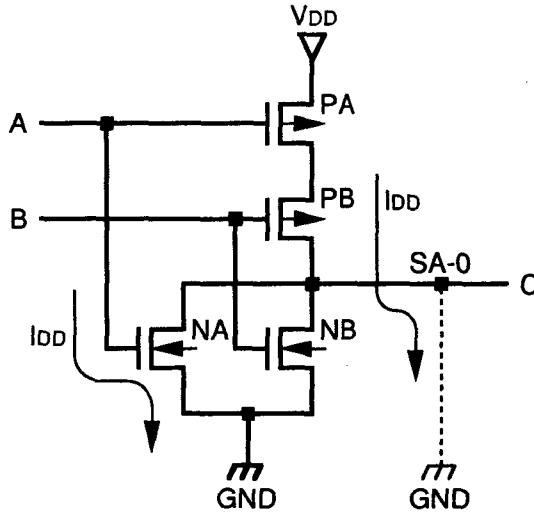
[例 5.1] 図 5.1 (a) に示す 2 入力 CMOS NOR ゲートの出力信号線 C の 0 縮退故障 SA-0 と、nMOS トランジスタ NA のゲート端子と基板 p-Well との間のゲート酸化膜短絡故障 GOSF とについて説明する。

信号線 C に 0 縮退故障が発生したとき、この信号線は何らかの物理的原因により GND に接続されていると見なすことができる⁽⁶³⁾。従って、C を論理値 '1' に設定すると、 V_{DD} から導通したトランジスタ PA, PB を通って GND 側へ電源電流 I_{DD} が流れる。

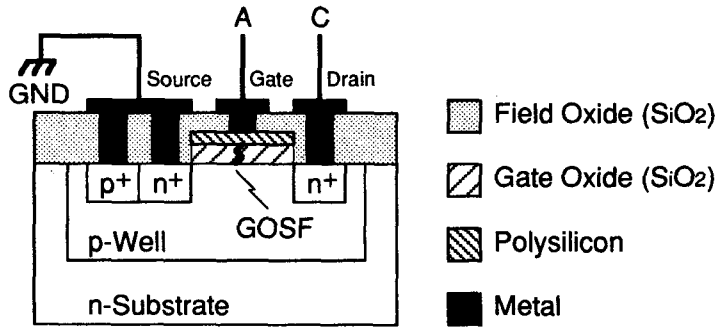
次に、図 5.1 (a) 中の nMOS トランジスタ NA のゲート端子と基板との間のゲート酸化膜短絡故障について考える (図 5.1 (b))。このトランジスタの基板 p-Well は GND に接続されているため、入力信号線 A は GND の電圧値になる。従って、A を論理値 '1' に設定す

ると、A から NA の基板側へ電源電流 I_{DD} が流れる。

以上のことから、縮退故障とゲート酸化膜短絡故障に対しては、各信号線に論理値 '0' と '1' とを割当ててすることで、電流テストにより故障検出が可能である。



(a) 縮退故障



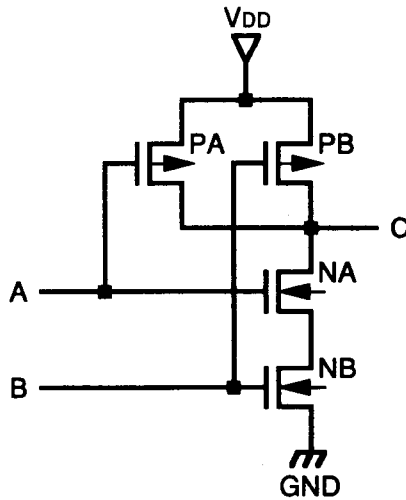
(b) ゲート酸化膜故障

図 5.1 故障モデル

[補題5.2] ゲート内の導通故障を検出するためのテストベクトルは、そのゲートの縮退故障を顕在化するテストベクトルと同一である。

(証明) 縮退故障を顕在化するテストベクトルは、正常な場合と故障の場合とでゲ-

ト出力値が異なる。導通故障は、そのトランジスタのゲート端子が常に一方の論理値に縮退していると考えられる。正常ならばこのトランジスタと、これと対をなすトランジスタとのどちらか一方のみが導通状態になるのに対し、導通故障ならばこれら両方のトランジスタが導通する。よって、縮退故障に対するテスト入力をゲートに印加したならば、電流テストで導通故障をテストできる。逆に、導通故障をテストするためには、ゲートの出力と、電圧源端子 V_{DD} , GND との間にあるすべての経路を少なくとも1回導通させなければならない。また、直列に接続されたトランジスタは1個ずつテストしなければならない。このような条件を満たす入力値はこのゲートの縮退故障を顕在化する入力である。従って、導通故障を検出するためのテストベクトルは、縮退故障を顕在化するテストベクトルと同一である。 (証明終)



(a) 2入力 CMOS NAND ゲート

A	B	SONF	GOSF(G-Sub.)	SAF
0	1	NA	NB, PA	A ₁ , B ₀ , C ₀
1	0	NB	NA, PB	A ₀ , B ₁ , C ₀
1	1	PA, PB	NA, NB	A ₀ , B ₀ , C ₁

(b) テストベクトルと検出故障

図 5.2 2入力 NAND ゲートのテストベクトル

[定理 5.1] ゲート内の導通故障に対するテストベクトルの印加により、電流テストで縮退故障、導通故障、ゲート酸化膜短絡故障の検出が可能である。

(証明) 補題 5.1, 5.2 より自明である。 (証明終)

定理 5.1 より、電流テストでは導通故障のみに対するテストを考える。

図 5.2 (a) の 2 入力 CMOS NAND ゲートのテストベクトルとその検出可能な故障との対応を図 5.2 (b) に示す。SONF, GOSF, SAF は、それぞれ導通故障、ゲート酸化膜短絡故障、縮退故障を表している。なお、ゲート酸化膜短絡故障は、ゲート端子と基板とが短絡したときに故障の検出が可能なトランジスタを表している。また、縮退故障については、各信号線の検出可能な縮退故障の縮退値を添字で表している。この図から明らかなように、2 入力 CMOS NAND ゲートに対し、3 つのテストベクトルで導通故障、ゲート酸化膜短絡故障、縮退故障をすべてテストできる。

なお、トランジスタの開放故障、断線故障、ブリッジ故障などについては本論文で述べる電流テストでは特に対象としないが、MOS トランジスタの開放故障やゲート端子が浮遊端子となるような断線故障の一部は電流テストでもテスト可能であるといわれている⁽⁶⁴⁾。

5.2 テスト用回路

5.2.1 機能設計

外部テスト装置による電流テストの動作速度、分解能の欠点を組込みテストにより補うためには、組込む回路に対して以下の機能が絶対的に要求される。

[要求 5.1] 動作速度が速い。

[要求 5.2] 分解能が高い。

また、被テスト回路内に組込む回路に対しては、以下の特性が要求される。

[要求 5.3] テスト用回路による電圧降下が小さい。

[要求 5.4] テスト用回路の構成が簡単である。

[要求 5.5] テスト用回路の面積が小さい。

[要求 5.6] 被テスト回路の通常動作での性能低下が小さい。

これまで提案されている電流テストは、外部テスト装置を用いて行われていたため要求 5.1, 5.2 の機能を満足できなかったが、組込み電流テストにすることで、ある程度これらの要求を満足することができる^(60,61)。

しかし、CMOS回路ではゲートのスイッチング時に動的電流が流れるため、異常な静的電流の瞬時値計測を高速に行うことはかなり困難である。これに対処するため、本論文では、単位時間内の電流の積分値を計測するテスト用回路を提案する。この回路は動的電流を含んだ電流計測を行い、前述の問題に対処している。

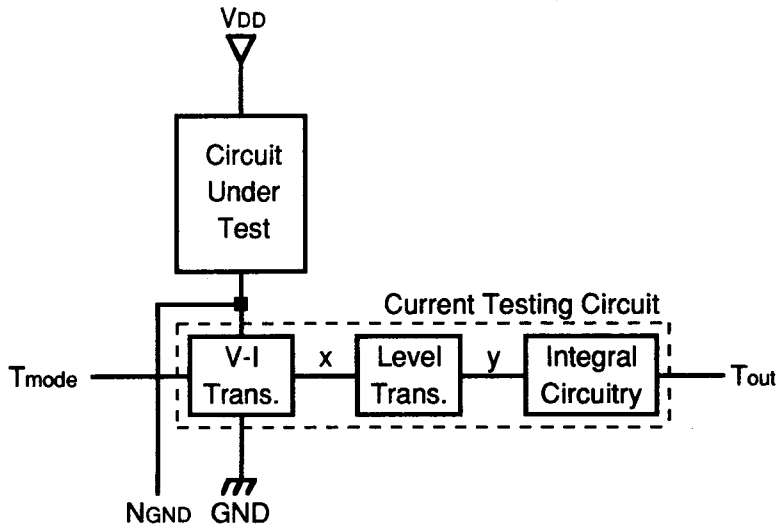


図 5.3 テスト用回路のブロック図

図 5.3 にテスト用回路のブロック図を、図 5.4 に動作概要をそれぞれ示す。テスト用回路は V-I 変換回路、レベル変換回路、積分回路から成っている。V-I 変換回路は被テスト回路内を流れる電流を電圧に変換する(図 5.4 (a))。要求 5.3 を満足するためには極力小さい値の抵抗を挿入しなければならない。レベル変換回路は、V-I 変換回路で発生させた電圧を適切な論理レベルに変換する。V-I 変換回路に基準値 I_s 以上の電流が流れ込んだとき、正常値と異なる論理値を出力する(図 5.4 (b))。すなわち、アナログ量をデジタルに変換する。積分回路は、故障を表す論理値がある一定時間以上継続したときに故障信号を出力する(図 5.4 (c))。図 5.4 (b) では基準時間間隔 t_s に対し、 $t_1 < t_s$ 、 $t_2 < t_s$ である。テスト用回路は正常時 $T_{out} = '1'$ 、故障時 $T_{out} = '0'$ を出力する。要求 5.6 を解決するために被テスト回路の GND 側に N_{GND} 端子を設けた。この端子はテスト時にはオープン状態であるが、通常動作時には GND に接続され、V-I 変換回路による電圧降下をなくし被テスト回路の性能低下を防止する。

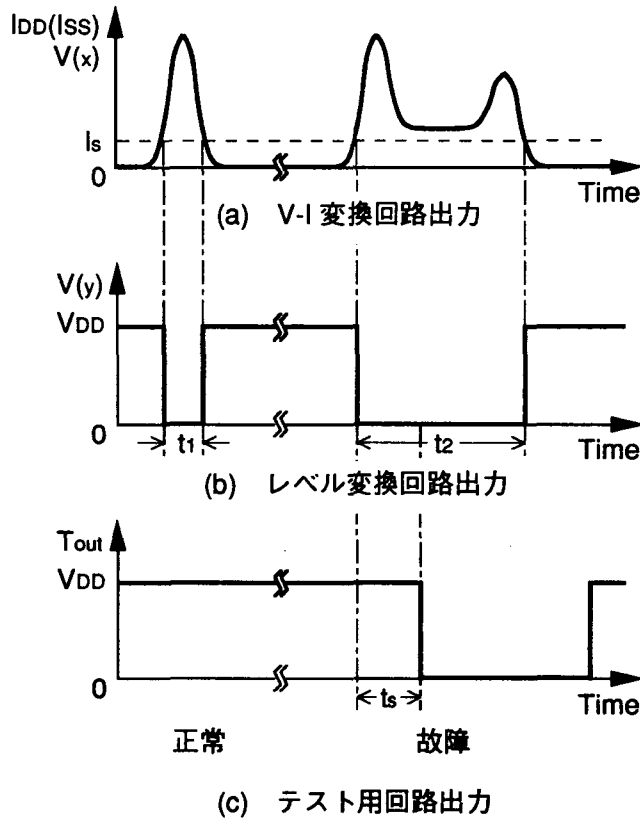


図 5.4 動作概要

5.2.2 回路設計

5.2.1項で述べた機能を満足するための各ブロックの回路設計を行う。回路設計は $2\mu\text{m}$ ルールを仮定している。nMOS, pMOSそれぞれのトランジスタのしきい値電圧は $V_m = |V_{\phi}| = 0.6\text{ V}$ に設定した。また、回路シミュレーションにはSPICE3 (Simulation Program with Integrated Circuit Emphasis, Version 3B.1)を使用し、 $V_{DD} = 5.0\text{ V}$ 、温度 $= 27^\circ\text{C}$ で実施した。本論文で使用するSPICE3は、1987年にカリフォルニア大学バークレー校からリリースされたバージョン3B.1である。図5.5に本論文で提案する電流テスト用回路の回路図を示す。以下に、回路の詳細についてブロックごとに説明する。なお、図中にトランジスタのチャンネル幅 $W [\mu\text{m}]$ /チャンネル長 $L [\mu\text{m}]$ 、抵抗値 $[\Omega]$ 、容量値 $[\text{F}]$ を記した。また、以下の説明では、節点 x での電圧を $V(x)$ で表すことにする。

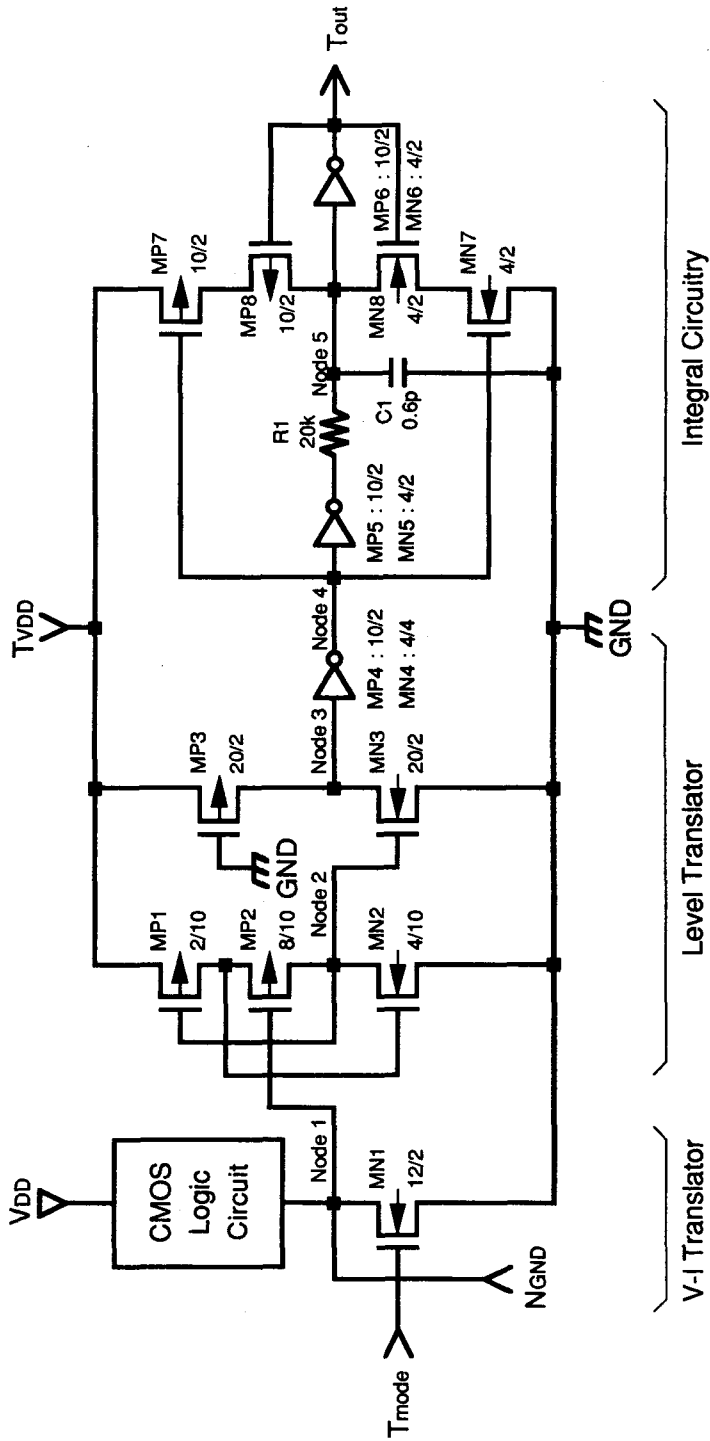


図 5.5 電流テスト用回路

A. V-I 変換回路

要求 5.1, 5.3 を満足するためには被テスト回路の電源端子に挿入する抵抗値は小さくしなければならず, 一方, 要求 5.2 を満足するためには抵抗値を大きくしなければならない. ここでは, 要求 5.1, 5.3 を優先させ, 要求 5.2 については次段のレベル変換回路で対応することにした. nMOS トランジスタの線形領域におけるオン抵抗を利用して V-I 変換を行っている. SPICE3 による $W/L = 12/2$ の nMOS トランジスタの V-I 特性を図 5.6 に示す. トランジスタのオン抵抗値 R_{on} は $I_{ds} = 1 \text{ mA}$ で約 $R_{on} = 1 \text{ k}\Omega$, $I_{ds} = 2 \text{ mA}$ で約 $R_{on} = 1.1 \text{ k}\Omega$ である. なお, トランジスタ MN1 のサイズ W/L は被テスト回路の動作速度に依存して決まる.

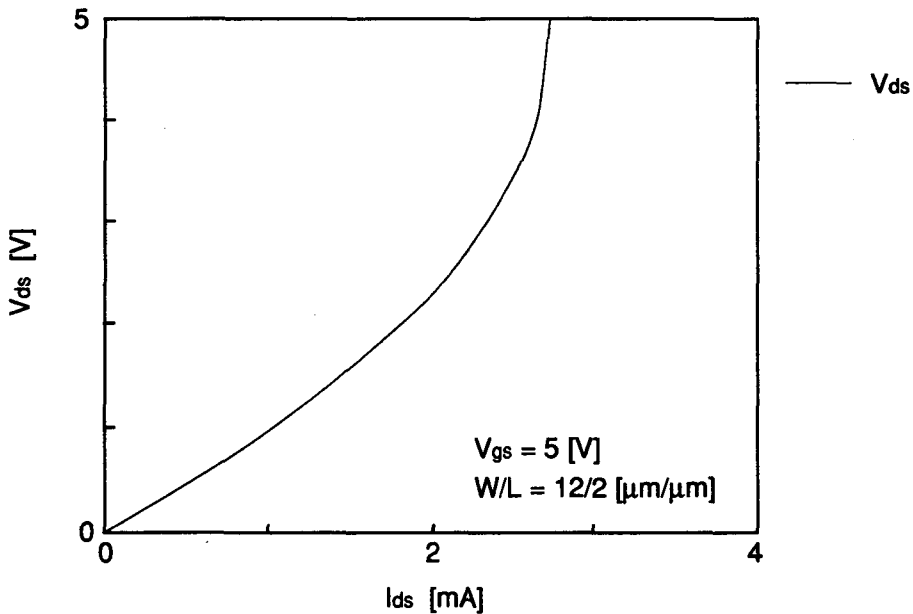
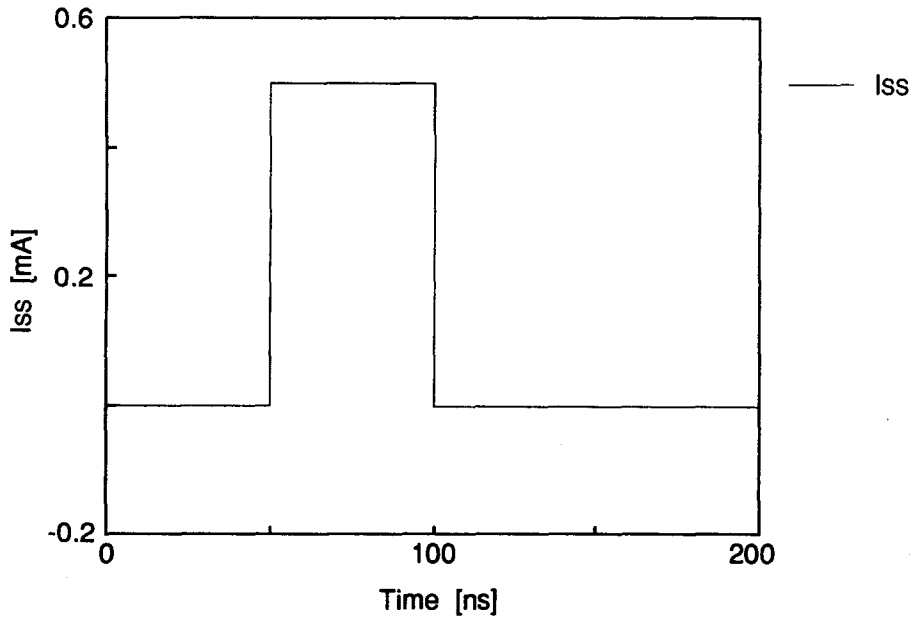


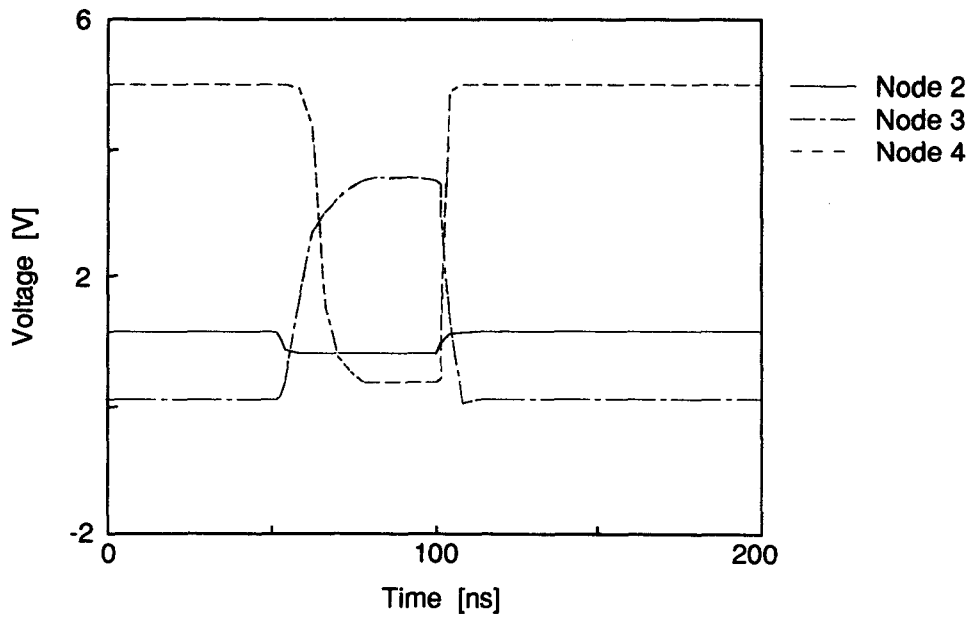
図 5.6 nMOS トランジスタの V-I 特性

B. レベル変換回路

レベル変換回路の目的はアナログ入力をデジタル出力に変換することである. Node 1 をトランジスタ MN3 に直接接続した場合, V_{in} 以上の電圧が発生しないと故障を表す論理値が次段へ出力せず要求 5.2 を満足できない. 故障判定の電流値を小さくするためには



(a) 印加電流



(b) シミュレーション結果

図 5.7 レベル変換回路のシミュレーション結果

MN1の抵抗値を大きくしなければならず、要求5.1, 5.3に反する。従って、MN3の前段にMP1, MP2, MN2によって構成される可変抵抗を挿入した。Node 2の電圧はNode 1に発生した電圧値により $V_m = 0.6\text{ V}$ を中心値として変化する。MP1, MP2, MN2の各サイズは要求する故障判定の基準電流値によって決まる。ここでは判定の基準電流値として、 $I_s = 0.5\text{ mA}$ となるように各トランジスタのサイズを決めている。 $I_{ss} > I_s (= 0.5\text{ mA})$ のとき $V(\text{Node } 2) < 0.6\text{ V}$ となり、トランジスタ MN3 はオフし、 $V(\text{Node } 3) = V_{DD}$ となる。INV1 は Node 4 のレベルを完全な論理レベルに変換するインバータであり、上記の場合 '0' を出力する。レベル変換回路の SPICE3 による回路シミュレーション結果を図 5.7 (a), (b) に示す。シミュレーションは電流印加型で行った。

C. 積分回路

積分回路は、故障を表す論理値 '0' がある時間以上連続して入力すると、テスト用回路としての最終的な故障判定値 $T_{out} = '0'$ を出力する。すなわち、故障電流 ($> 0.5\text{ mA}$) が時間 t_s 以上継続して流れたとき $T_{out} = '0'$ となる。この動作を実現するために CR 積分回路を用いている。 t_s の最大値は被テスト回路の1クロック周期分の時間である。インバータ INV2,

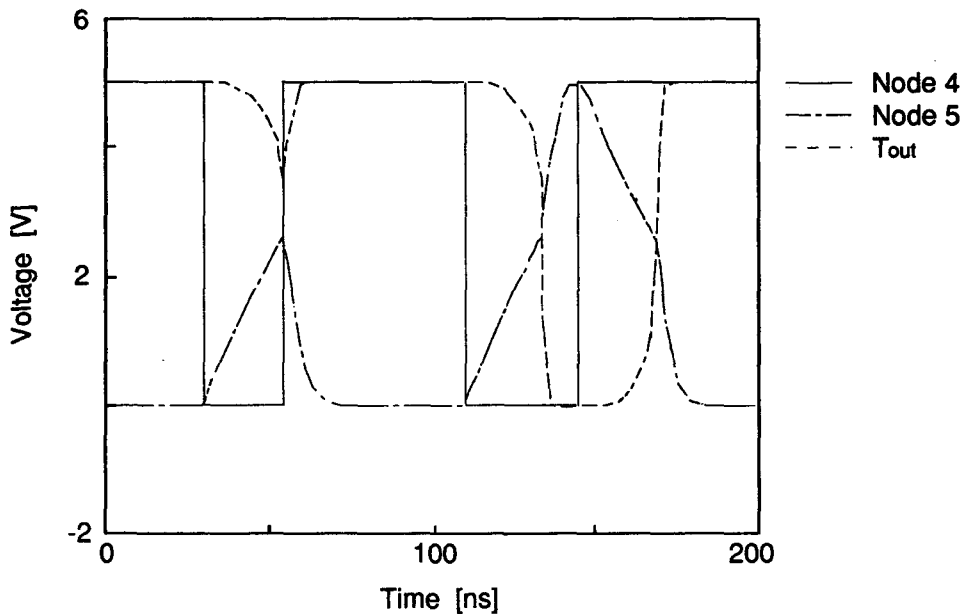


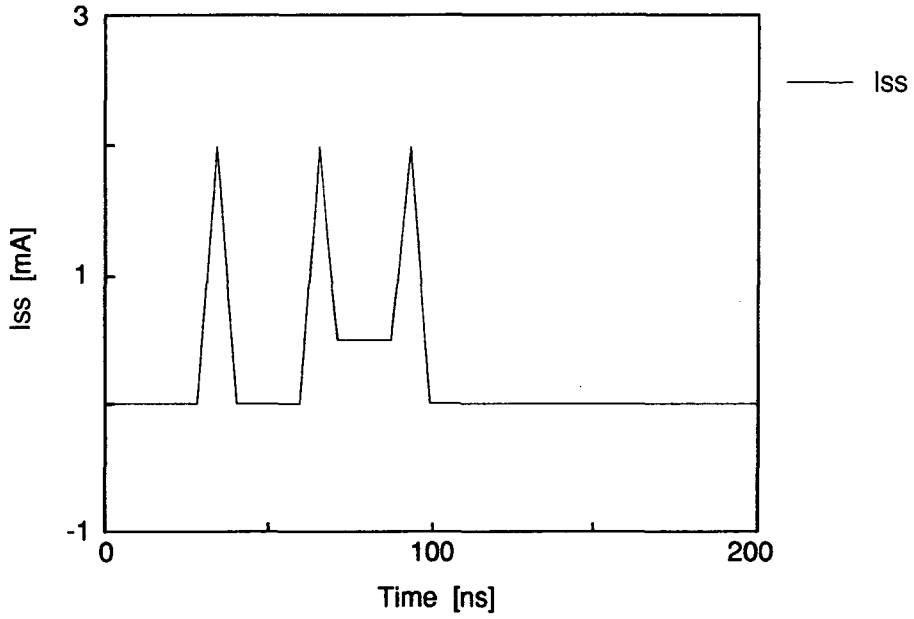
図 5.8 積分回路のシミュレーション結果

INV3 は、それぞれのしきい値電圧が $1/2V_{DD}$ となるような nMOS トランジスタと pMOS トランジスタとのサイズ比に設定している。入力論理値が変化するとき、Node 5 は V_{DD} と GND とから切離された状態となり C1, R1 の時定数 τ によって決まる時間で充放電する。従って、変化した入力論理値 $V(\text{Node 4})$ が時間 τ 以上継続して一定ならば $V(\text{Node 5}) > 1/2V_{DD}$ となり T_{out} は入力論理値によって決まる論理値に変化する。C1, R1 の各値は故障判定の基準時間 t_s によって決まるが、 $\tau \leq t_s$ であることから C1, R1 の値は被テスト回路の動作速度にも依存する。被テスト回路の動作速度を 30 MHz とした場合の積分回路の SPICE3 による回路シミュレーション結果を図 5.8 に示す。なお、図 5.5 の回路では、 $I_{SS} = 0.5 \text{ mA}$ が 33 ns (30 MHz) 以上連続して流れたとき故障であると判断できるようにするために $C1 = 0.6 \text{ pF}$, $R1 = 20 \text{ k}\Omega$ としたが、これはレベル変換回路自身に遅延があるため、被テスト回路の動作速度よりも小さい時定数となっている。

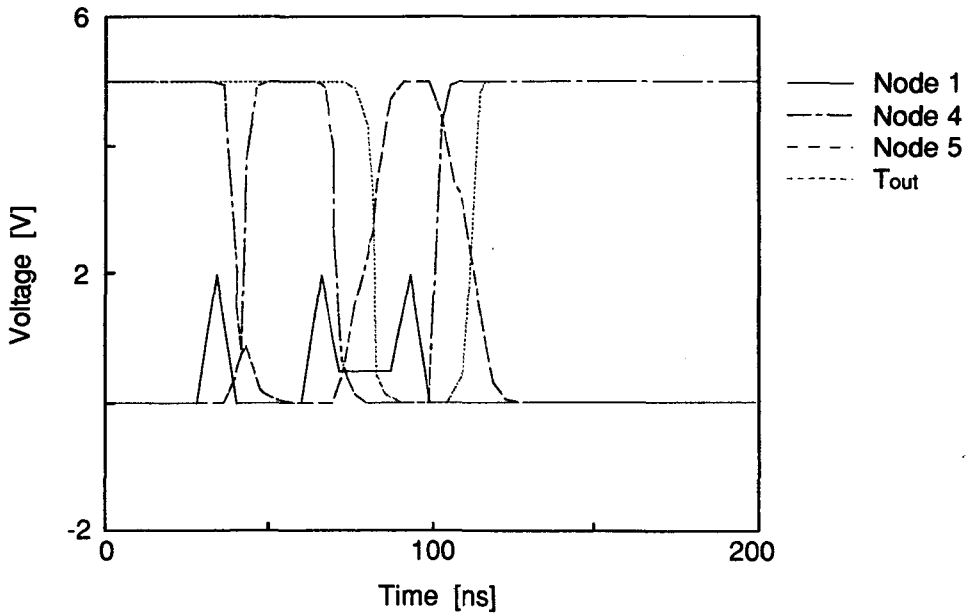
D. 回路動作

図 5.5 に示したテスト用回路全体の動作について述べる。図 5.9 に SPICE プログラムを、図 5.10 (a), (b) にその回路シミュレーション結果を示す。通常動作では $T_{mode} = '0'$, $N_{GND} = \text{GND}$, $T_{VDD} = \text{GND}$ であり、テスト動作では $T_{mode} = '1'$, $N_{GND} = \text{Open}$, $T_{VDD} = V_{DD}$ である。正常時には $T_{out} = '1'$ 、故障時には $T_{out} = '0'$ となる。

正常な CMOS 回路は動的電流のみが生じ、静的電流はほとんど流れない。静的電流が流れないならば $V(\text{Node 1}) = \text{GND}$, $MN3 = \text{On}$, $V(\text{Node 3}) = \text{GND}$ となり、 $T_{out} = '1'$ である。動的電流が流れたとき $V(\text{Node 2}) < V_m$ ならば $V(\text{Node 3}) = V_{DD}$, $\text{Node 4} = '0'$ となる。しかし、 $\text{Node 4} = '0'$ となっている時間が故障判定の基準時間間隔 t_s より短ければ Node 5 の容量を十分に充電できず、 $V(\text{Node 5}) < 1/2V_{DD}$ のままであり、よって、 $T_{out} = '1'$ となりテスト用回路は被テスト回路が正常であると判断する。逆に、被テスト回路内に故障が存在し、かつ、顕在化したとき、異常な静的電流が流れる。このときの静的電流値が故障判定の基準電流値 I_s 以上ならば $V(\text{Node 2}) < V_m$ となり、 $\text{Node 4} = '0'$ となる。 t_s は被テスト回路の 1 クロック周期分に設定されており、故障による電流が t_s 以上継続して流れるため $V(\text{Node 5}) > 1/2V_{DD}$ となり、 $T_{out} = '0'$ を出力する。よって、テスト用回路は被テスト回路が故障であると判断する。



(a) 印加電流

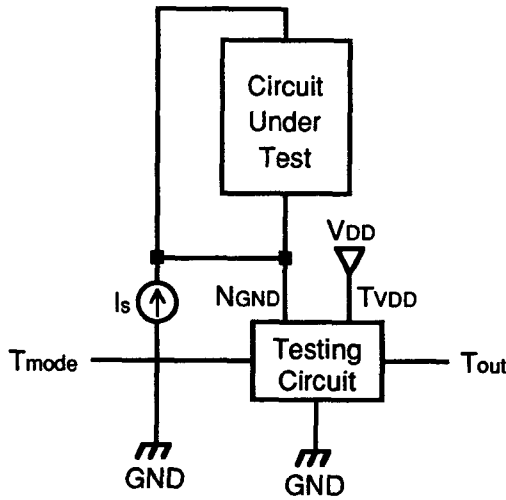


(b) シミュレーション結果

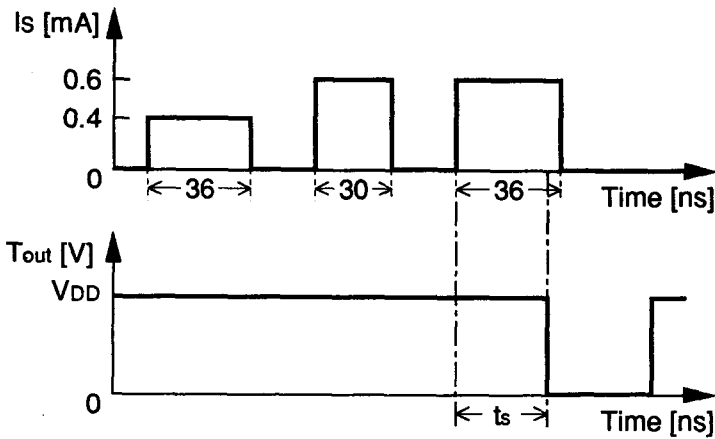
図 5.10 テスト用回路のシミュレーション結果

5.2.3 テスト用回路の自己テスト

被テスト回路内に組込まれるテスト用回路自身のテスト，自己テスト方法について述べる。テスト用回路は，入力がアナログ，出力がデジタルのアナログ回路であるため，トランジスタの導通故障，開放故障などの故障モデルでは故障を表現することが困難であると考えられる。よって，回路機能を確認める機能テストにより自己テストを行う。



(a) 測定回路のブロック図



(b) 印加波形と期待出力値

図 5.11 自己テスト方法

テスト用回路の機能は、トランジスタ MN1 を流れる電流値が基準電流 I_s 以上で、かつ、それが基準時間 t_s 以上連続して流れたとき出力 $T_{out} = '0'$ となり、これ以外るとき $T_{out} = '1'$ となることである。従って、 N_{GND} 端子への電流、または、電圧印加による AC テストでテスト可能である。図 5.11 (a) に電流印加型の測定回路を、図 5.11 (b) にその入出力例を示す。また、レベル変換回路のトランジスタは抵抗の役割を果たしているため、これらの概略値を確かめるためにテスト用回路の T_{VDD} 端子での電流計測を行う。以上のことを行うことで自己テストが可能である。

5.2.4 レイアウト設計

レイアウト設計とは、実際に回路をシリコンウェハ上に形成するためのある幾何学的パターンを作成することである。この幾何学的形状の幅、間隔、延長、重なりなどの制約条件を記したものが設計ルールである。実際の設計ルールは使用するプロセスによって異なるが、Mead と Conway は基本距離単位 λ を用いて無次元の相対的な量で表現した設計ルールを提案した⁽⁸⁵⁾。この設計ルールは λ ベースの設計ルールと呼ばれている。彼らの設計ルールは nMOS 回路に対するものであるので、本論文では、これを CMOS 回路に拡張した Weste と Eshraghian の λ ベースの p ウェルルール^(86,87) を用いてレイアウト設計を行った。

図 5.5 のテスト用回路のマスキレイアウトを図 5.12 に示す。このレイアウトは p ウェルプロセスを使用した最小寸法が $2\mu\text{m}$ である $2\mu\text{m}$ ルール ($\lambda = 1\mu\text{m}$) を想定している。コンデンサはゲート容量 C_g を、抵抗はポリシリコンのシート抵抗 R_s をそれぞれ利用し実現している。各々の単位当りの値は $C_g = 4 \times 10^{-4} \text{ pF}/\mu\text{m}^2$ 、 $R_s = 100 \Omega/\square$ ^(86,87) を使用した。回路のブロックサイズは $134 \times 89 \mu\text{m}^2$ である。

5.3 テスト生成

5.3.1 テスト系列と回路動作

あるテスト系列を正常な被テスト回路に印加したとき、図 5.13 (a) の太線に示すような電流が流れたとする。このとき、細線は図 5.5 の回路の Node 5 での電圧値である。ここで、図中の各値は定量的に表している。また、 T_i ($i = 0, 1, \dots, 4$) はテストベクトルを、 t_j ($j = 1, 2, 3, 4$) は積分区間をそれぞれ表す。このとき、C1, R1 の各値は、 t_s 時間継続して I_s

* Ω/\square は 1 平方当りの抵抗値を表す。

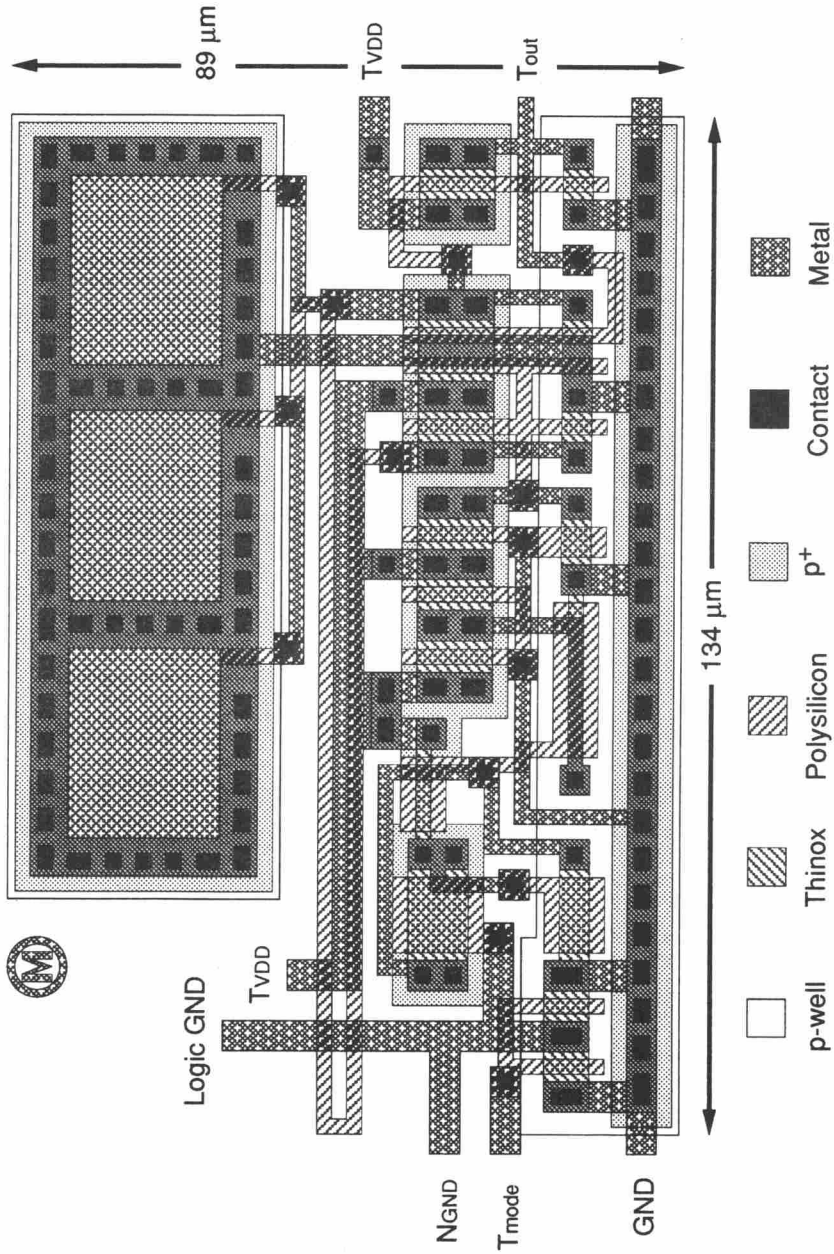


図 5.12 マスクレイアウト

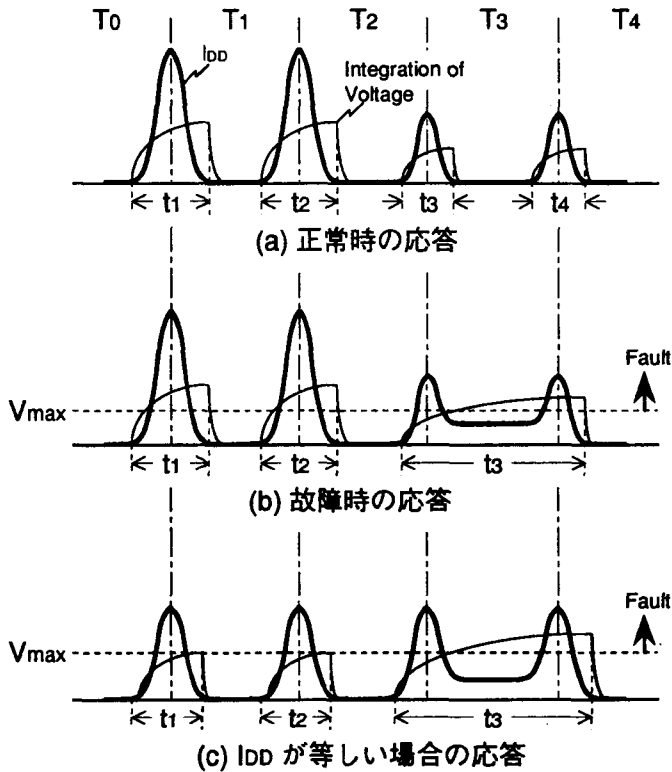
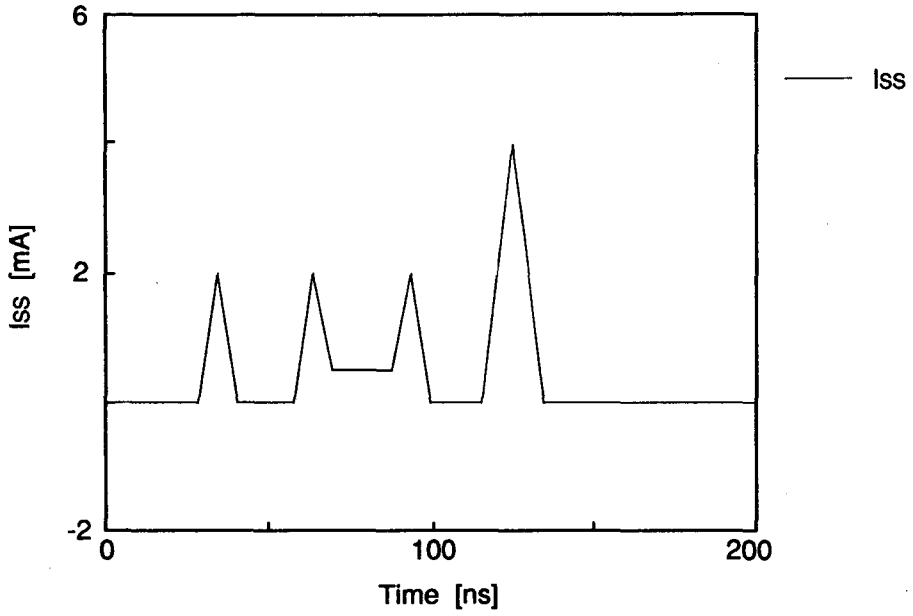
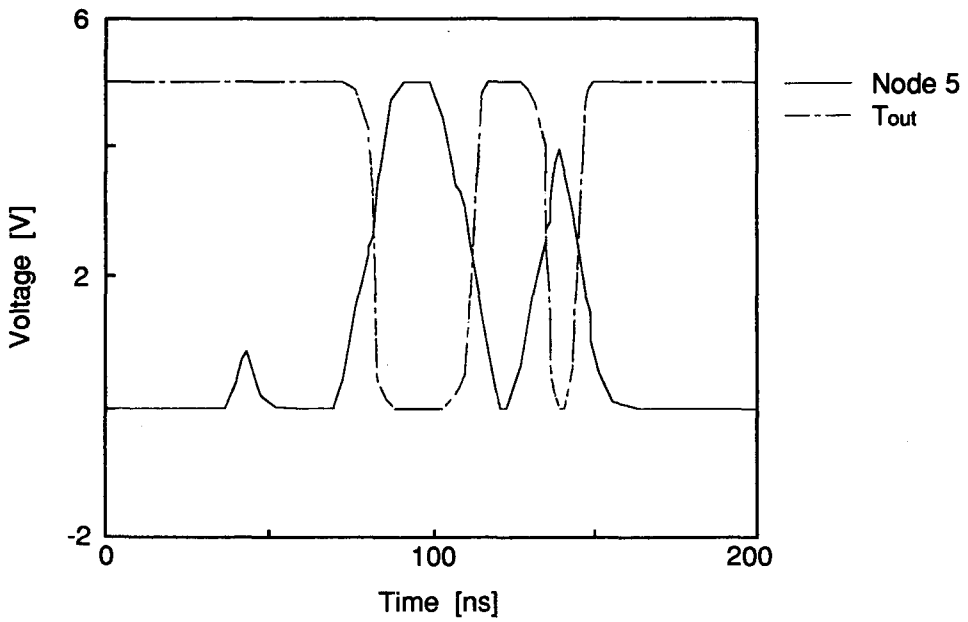


図 5.13 回路応答例

以上の電流が流れると故障であると判断できるように設定しているが、それと同時に、Node 5 での電圧の積分の最大値 V_{max} は $T_0 - T_1$ 間、 $T_1 - T_2$ 間で得られるので、 $V_{max} < 1/2V_{DD}$ を満たすように $C1$, $R1$ の各値が設定されている。今、図 5.13 (b) に示したようにテストベクトル T_3 を印加したとき故障が顕在化され、静的電流が流れたとする。このとき、 t_3 での積分電圧値は $1/2V_{DD}$ 以上になるように $C1$, $R1$ の各値を決定しなければならない。しかし、 $T_0 - T_1$ 間、 $T_1 - T_2$ 間での動的電流値が大きいため Node 2 の電圧が $I_s = 0.5 \text{ mA}$ の場合より十分に低くなり、その結果、レベル変換回路の遅延時間が小さくなって $V(\text{Node } 5) > 1/2V_{DD}$ となる。これは各テストベクトル間での動的電流値が異なることに起因している。図 5.13 (c) に示したように各テストベクトル間での動的電流値が等しければ、正常時の V_{max} 値は全積分区間で等しくなり、正常/故障の判定が容易になる。上述した場合の回路シミュレーションの結果を図 5.14 (a), (b) に示す。次項では、動的電流値が等しくなるようなテスト系列の生成法について述べる。



(a) 印加電流



(b) シミュレーション結果

図 5.14 図 5.13 のシミュレーション結果

5.3.2 テスト系列生成手法

テストベクトル間での動的電流値が等しくなるようなテスト系列生成法，すなわち，回路内のゲート出力が変化するゲート数が，ほぼ等しくなるようなテスト系列の生成法について述べる．このようなテスト系列を生成するために，本論文では，初めに目標となる変化数を設定し，次にこれを満足するようなテスト系列の生成を行う．

実際には，動的電流は，ゲートの出力値の変化方向やゲートの駆動能力などによって異なるが，本論文では以下のことを仮定する．

[仮定 5.1] ゲート出力値の '0' → '1' の変化と '1' → '0' の変化とでは流れる動的電流は等しい．また，すべてのゲートの駆動能力は等しい．さらに，回路の内部遅延やハザードはないものとする．

表 5.1 擬似乱数 500 ベクトル印加時のゲート出力の変化数

回路名	ゲート	MOSFET	故障	変化数/2ベクトル			
				平均	最大	最小	標準偏差
adder	76	214	183	32	52	0	10
74181	121	470	356	41	69	0	11
cm880	555	1802	1456	172	289	83	30
cm1355	636	2308	1790	208	269	130	24
cm1908	1097	3430	2812	436	562	190	61
cm2670	1799	5364	4481	660	878	433	81
cm3540	2482	7504	6234	794	1077	304	121

被テスト回路に擬似乱数系列を印加したときのゲート出力の変化数を目標変化数として設定する．加算器 adder，算術論理演算器 74181，ISCAS '85 のベンチマーク回路に対し，擬似乱数 500 ベクトルの系列を印加したときのゲート出力が変化したゲート数を表 5.1 に示す．各回路は NAND，NOR，NOT の各ゲートのみで再構成した回路である．ゲートは回路内の全ゲート数を，MOSFET はその全トランジスタ数を表している．故障は回路内の導通故障の総数を表している．なお，同一ゲート内で並列に接続されたトランジスタの導通故障は，同一のテストベクトルで検出され区別できない．よって，これらの故障を等価と見なし故障数は 1 個として数えている．最後の 4 つの欄は，2 ベクトル間で，故障がない場合に出力が変化したゲート数の平均値，最大値，最小値，標準偏差をそれぞれ表して

いる。表 5.1 の結果から、平均の変化数はゲート数の約 30%、標準偏差はその約 10 % であることから、2つのテストベクトル間でのゲート出力の目標変化数は以下のように与えられる。

$$\text{目標変化数} = (\text{ゲート数} \times 0.3) \pm (\text{ゲート数} \times 0.03) \quad (5.1)$$

ここで、用語の定義を与える。

[定義 5.3] 信号線 s のレベルは、外部入力信号線から s までの最多のゲート段数として定義する。すなわち、 s のレベル $l(s)$ は、

$$l(s) = \begin{cases} 0 & (s \text{ が外部入力信号線の場合}) \\ 1 + \max \{l(i_1), l(i_2), \dots, l(i_n)\} \end{cases} \quad (5.2)$$

と再帰的に定義する。ここで、 $i_j (j=1, 2, \dots, n)$ は、信号線 s を出力とするゲートの入力信号線を表す。

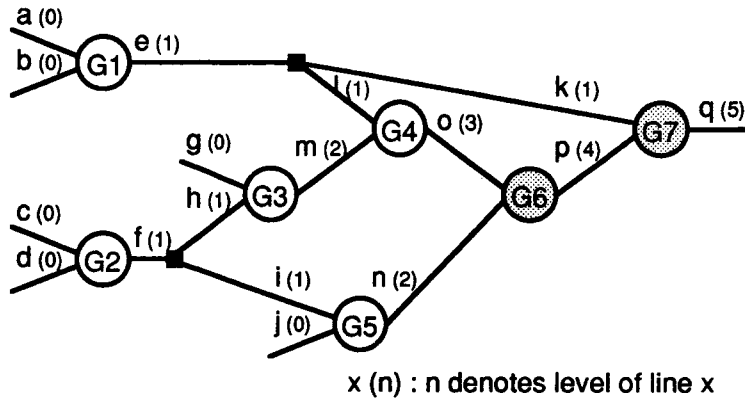
[定義 5.4] 経路 P 上にあるゲート G において、 G の P 上にない入力 i をオフ経路入力と呼ぶ。

[定義 5.5] ある論理値が回路の外部入力端子に割当てられている状態で、ゲートの 1 本の入力信号線の論理値の変化がそのゲートの出力へ伝搬する経路を伝搬可能経路と呼ぶ。また、このゲートのそれ以外の経路を伝搬不能経路と呼ぶ。

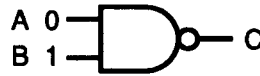
[定義 5.6] 少なくとも 1 つの伝搬可能経路を持つゲートを伝搬可能ゲートと呼ぶ。また、これ以外のゲートを伝搬阻止ゲートと呼ぶ。

[例 5.2] 図 5.15 (a) の回路において、() 内の数値が信号線のレベルを表す。また、ゲート $G1 \sim G5$ は非再収斂ゲートであり、ゲート $G6, G7$ は再収斂ゲートである。経路 $a-e-l-o$ に対し、入力 b, m がオフ経路入力である。また、図 5.15 (b) の 2 入力 NAND ゲートに $(A, B) = (0, 1)$ が印加されているとき、このゲートは伝搬可能ゲートであり、また、 $A-C$ が伝搬可能経路、 $B-C$ が伝搬不能経路である。すなわち、入力 A の論理値の変化は直ちに出力 C へ伝搬するが、入力 B の論理値の変化は A の論理値を変化させない限り C へ伝搬しない。

通常、縮退故障のテストでは、故障の顕在化と外部出力信号線への故障の影響の伝搬とから成っている。一方、電流テストでは、その故障の検出方法から故障の顕在化のみを行



(a) 回路例 1



(b) 2入力 NAND ゲート

図 5.15 用語の説明図

うだけで十分である。これは、縮退故障のテストにおける故障の顕在化のみを使用して実現可能であるが、ゲートの出力値の変化数を見積ることが困難であると考えられる。よって、本論文では D アルゴリズムの一致操作と PODEM アルゴリズムの後方追跡とを組合せた後方操作を利用したテスト系列の生成を行う。

初めに、対象となる故障の存在するゲートの入力にその故障を顕在化する入力値を割当てる。次に、この入力信号線から外部入力信号線へ向かって1本の経路を設定する。このとき、ゲートを通過するごとに、そのゲートの入力値を一致操作によって決定する。ゲートへの入力値としては、そのゲートを伝搬可能ゲートとする入力値を割当てる。設定可能な経路、および、割当て可能なゲートの入力値が複数ある場合は、ゲートの出力値の変化数が多くなると予想されるものを選択する。その選択方法については後述する。一致操作を外部入力信号線まで行った時点では、オフ経路入力は未正当化信号線として残っている。これらの信号線を正当化するために、割当てられている論理値をそれぞれ目標値として後方追跡を行い、外部入力値を決定する。このとき、目標値となる未正当化信号線の論理値の間に矛盾があり、テストベクトルの生成に失敗した場合、対象故障の位置から他の

設定可能な経路に沿って同様の操作を繰り返す。

[例 5.3] 図 5.15 (a) の回路のゲート G6 の入力 n に接続されたトランジスタの導通故障を考える。初めに、G6 の入力 o, n に対象故障を顕在化する入力値を割当てて、ゲート G4 の入力の変化数とゲート G5 のそれとを比較して、G4 の入力の変化数が多いならば G4 の入力信号線 l, m に対し一致操作を行う。同様にゲート G3, G2 に対し一致操作を行い、外部入力信号線 c, d の論理値が決定される。ここで、設定された経路は o-m-h-f-c (d) である。次に、未正当化信号線 n, l に対し後方追跡を行い、外部入力信号線 a, b, j の論理値を決定する。これらの外部入力信号線の論理値は、PODEM アルゴリズムと同様の後方追跡操作により決定される。

未検出故障のうち、ゲート端子への入力信号線のレベルの高いものから順次テストベクトルを生成する。また、対象故障の位置から外部入力信号線まで一致操作を行う際、テスト生成が成功したときのおおよそその変化数を見積る。ここでは、設定した経路に対するオフ経路入力信号線のレベルの算術和と設定した経路内に含まれる信号線数との総和を予想変化数として見積っている。設定可能な経路、および、割当て可能な入力値が複数存在する場合、以下の規準に従って選択する。

[規準 5.1] テスト生成が成功したとき、より多くの変化が起こると予想される経路、および、入力値を選択する。

[規準 5.2] 再収斂ゲート数の少ない経路を選択する。

[規準 5.3] 未検出故障を多く含む経路を選択する。

次に、規準 5.1 ~ 5.3 の設定理由について述べる。

規準 5.1 において、変化数が多くなるような経路、入力値を選択する理由は、目標変化数を回路内のゲート数の約 30% と設定したので、変化数が多くなるように選択した場合の方が、テスト生成時により目標数に到達しやすいと考えているためである。

[定理 5.2] ある故障に対するテスト生成において、設定した経路内に、直前に生成されたテストベクトルによる伝搬不能経路を持つとき、伝搬不能経路を持つゲートに対し一致操作を行い、この経路を伝搬可能経路にすることができれば、そのゲートのオフ経路入力につながる少なくとも 1 本の経路上の論理値は変化する。

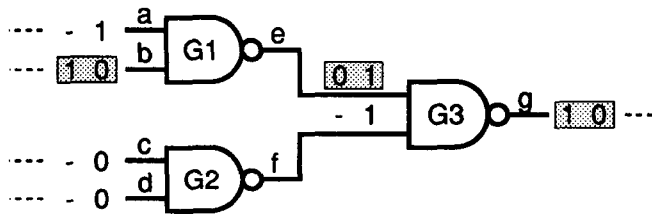
(証明) 伝搬不能経路上の各ゲートの少なくとも 1 本のオフ経路入力には入力値の変化の伝搬を阻止する論理値が印加されている。よって、この経路を伝搬可能経路にするた

めにはオフ経路入力に対し，入力値の変化をそのゲート出力に伝搬できる論理値を新たに印加しなければならない。 (証明終)

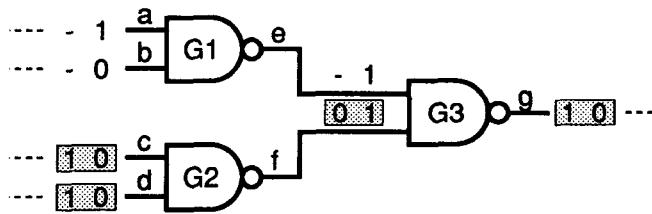
以上の定理より，対象故障の位置から外部入力信号線までの経路を設定するときの規準 5.1 に対する経路選択の規則を与える。

- [規則 5.1] 伝搬阻止ゲートの出力信号線を選択する。
- [規則 5.2] ゲートの出力値が変化する信号線を選択する。
- [規則 5.3] 伝搬不能経路である信号線を選択する。
- [規則 5.4] レベル番号の大きい信号線を選択する。

なお，規則 5.1 ~ 5.4 は，上から順にそれぞれの番号に対応した優先順位を持つ。



(a) 変化数が最小の場合



(b) 変化数が多い場合

図 5.16 回路例 2

[例 5.4] 図 5.16 の回路において信号線 g に論理値 '1' を設定する場合を考える。ここで，G1, G3 は伝搬可能ゲート，G2 は伝搬阻止ゲートである。また，b-e, e-g, f-g は伝搬可能経路，a-e, c-f, d-f は伝搬不能経路である。

(1) ゲート出力値の変化数が最小の場合 (図 5.16 (a))： 経路 g-e-a に沿って一致操作を行ったとする。このとき，a-e, e-g は共に伝搬可能な経路なので，論理値の変化はこ

の経路上でしか起こらない。よって、変化数は最小である。

(2) ゲート出力値の変化数が多い場合 (図 5.16 (b)): 経路 $g-f-c$ に沿って一致操作を行ったとする。ゲート G2 は伝搬阻止ゲート、すなわち、経路 $c-f$, $d-f$ は共に伝搬不能経路なので、G2 を伝搬可能ゲートにするためにはオフ経路入力 d の論理値を変化させなければならない。すなわち、外部入力信号線から d までの少なくとも 1 本の経路上の論理値を変化させなければならない。なお、 $l(d) > l(c)$ ならば経路としては $g-f-d$ を選択する。

この例で明らかなように、規則 5.1 ~ 5.4 に従って経路を選択したならば、より多くのゲート出力の変化を期待でき、かつ、その数を見積ることができる。

規準 5.2 は設定した経路に対しテストベクトル生成の成功確率を高くするために用いている。また、規準 5.3 はテストベクトル数を少なくするために用いている。なお、規準 5.3 に関し、次の定理を与える。

[定理 5.3] あるテストベクトルを回路に印加したとき、同時に伝搬可能ゲートとなるゲートは同時にテスト可能である。

(証明) 導通故障に対するゲートのテストベクトルと、そのゲートを伝搬可能ゲートとするゲートへの入力値が等しいことから明らかである。 (証明終)

一致操作で割当てるゲートの入力値において、そのゲートを伝搬可能ゲートとする入力値が必ずしも変化数を多くするものではない。しかし、定理 5.3 より伝搬可能ゲートとする入力をゲートに割当てた場合の方がテスト系列が短くなる可能性が高い。

5.3.3 テスト生成アルゴリズム

テスト生成アルゴリズムについて述べる。本論文で用いたテスト生成プログラムは、PODEM アルゴリズムに基づいた縮退故障のテスト生成プログラムの後方操作を使用している。後方操作には、PODEM アルゴリズムの後方追跡のほかに、D アルゴリズムの一致操作を追加した。D 設定操作、D 伝搬操作の部分は削除した。故障シミュレータは導通故障をシミュレートできるように変更した。このプログラムは C 言語を用い、SUN4 ワークステーション上に実現した。

図 5.17 にテスト系列生成のアルゴリズムを示す。前処理では故障リストの作成、回路のレベル付けを行っている。対象となる未検出故障に対し外部入力信号線までの経路に沿っ

```

procedure TPG-CT
  /* TPG with counting the number of transitions */
  begin
    pre-processing;
    val = number of objective transitions;
    initialize all signal lines;
    provoke fault simulation;
    while given faults exist do begin
      clear all signal values;
      count = 0;
      provoke consistency operation along a selected path;
      while count < val do begin
        generate a vector overlapping present vector;
        /* provoke backtrace for unjustified lines */
        if TPG is success then begin
          count += number of transitions;
          if count == val then
            break;
          if count > val then
            cancel present path;
            select another path;
          end
        else begin
          cancel present path;
          select another path;
        end
      end
      provoke fault simulation;
    end
  end;

```

図 5.17 テスト生成アルゴリズム

て一致操作を行う。その後、未正当化信号線に対し後方追跡を行い外部入力値を決定する。このとき、テスト生成が成功しても目標変化数に達しない場合は、さらに、このテストベクトルに重複させて他の未検出故障に対しても同様の操作を行う。逆に、変化数が目標値を越えた場合は最後に選択した故障に対する外部入力値を取消し、他の選択可能な経路に対してテスト生成を行う。目標変化数に達したとき、論理値が未決定の外部入力信号線には直前に生成したテストベクトルと同一の論理値を割当て、変化数の変動を防止している。任意の故障に対し、10回続けて選択した経路が適当でないためにテスト生成を失敗した場合、この故障に対するテスト生成を一旦打ち切り、他の故障についてテスト生成を行

う。なお、テスト生成には故障シミュレータを併用している。任意の故障に対するテストベクトルが生成されたとき、このテストベクトルで検出可能な他の未検出故障を故障シミュレータで求めている。従って、一旦テスト生成を打切った故障が、他の故障のテストベクトルによる故障シミュレーションで検出されたならば、その故障は検出できたことになる。

5.3.4 実験結果

表5.2にテスト系列の生成結果を示す。目標変化の欄は2つのテストベクトル間でのゲート出力の目標変化数を表している。系列長の欄は生成されたテスト系列長、すなわち、テストベクトル数を表している。テスト不能故障はテストベクトルの存在しない故障を表す。未検出故障は、バックトラック10回でテスト生成を打切った故障と10回続けてテスト生成を失敗し、かつ、最後まで残った故障とを表す。また、最後の欄は、本論文で作成したテスト生成プログラムの原型となったテスト生成プログラムによって生成された縮退故障のためのテスト生成の結果を表す。故障検出率 ρ は次式により算出した。

$$\rho = (1 - (\text{テスト不能故障数} + \text{未検出故障数}) / \text{故障数}) \times 100 \quad (5.3)$$

すべての回路に対し、生成されたテスト系列は目標の変化数を満足した。すなわち、すべてのテストベクトル間での動的電流値がほぼ等しいテスト系列を生成できた。さらに、加

表 5.2 テスト生成の実験結果

回路名	目標変化	系列長	テスト不能故障	未検出故障	検出率 [%]	縮退故障のベクトル*
adder	23±2	9	0	0	100.0	9
74181	36±4	52	4	10	96.1	88
cm880	167±17	47	0	0	100.0	86
cm1355	191±19	63	0	137	92.3	137
cm1908	329±33	72	0	79	97.2	165
cm2670	540±54	41	7	258	94.1	478
cm3540	745±74	36	23	496	93.1	260

*: 原形のテスト生成プログラムによる縮退故障に対するテスト生成の結果

算器 adder を除いて、従来の縮退故障に対するテストベクトル数に比べ、生成されたテストベクトルの数は少ない。すなわち、電流テストでは、従来の縮退故障に対するテストよりも少ないテストベクトル数で縮退故障のテストが可能である。

5.4 総括

本章では、組込み電流テストのためのテスト用回路の提案とその回路に適したテスト生成について論じた。CMOS 回路において本質的に生じる動的電流の影響を少なくするために、提案したテスト用回路は動的電流を含んだ電流計測、すなわち、単位時間内の電流の積分値を計測し、その値により正常/故障の判定をする。この計測方式により比較的分解能が高く、かつ、高速の電流テストが行える。SPICE3 シミュレータで回路シミュレーションを行い、要求した機能を満足する回路であることを確認した。さらに、テスト用回路の構成は比較的簡単で、かつ、その自己テストも可能である。レイアウト設計を行った結果、約 $134 \times 89 \mu\text{m}^2$ の面積で提案した回路を実現できることも確認できた。1 mm² のチップに対して面積オーバーヘッドは約 1.2% であり、提案した回路はかなり小さいことがわかる。また、提案した回路に適したテスト系列の生成手法について検討した。これは、テスト系列中のすべてのテストベクトル間での動的電流値をほとんど等しくするために、任意の連続する2つのテストベクトル間でのゲート出力値の変化数がほぼ等しいテスト系列を生成する。このテスト系列は従来の縮退故障に対するテスト生成プログラムの一部を利用して容易に生成でき、また、従来の縮退故障に対するテストベクトル数の 9 ~ 60% のテストベクトル数となった。

今後の課題としては、実際の被テスト回路に即した正確な回路のモデル化を行い、被テスト回路とテスト用回路とを統合化した回路シミュレーションを実施し、さらにテスト用回路の実用性について検討をする必要がある。テスト生成においては、変化数のみを考慮した経路設定以外に、テスト生成を成功しやすくするための経路の設定方法について検討を要する。また、ゲート出力の変化方向やゲートの駆動能力、被テスト回路の回路遅延やハザードを考慮したテスト生成についても検討の余地がある。

第 6 章

結論

集積回路が大規模化，高機能化するにつれ，テスト生成の困難さ，テストに要する時間が増大し，テストコストが高くなってきている．また，CMOS IC が集積回路の主流になり，従来の縮退故障モデルでは対応できない故障が現れるようになってきた．本論文ではテスト容易化設計手法を取入れ，メモリのテスト時間の短縮，および，CMOS 組合せ回路のゲートレベルとトランジスタ・レベルとの故障検出の問題を解決する新しいテスト方式について検討した．

メモリについては，1ワード線上のすべてのセルを同時にテストできる組込みテスト方式を提案した．CMOS 組合せ回路については，トランジスタの開放故障に対し，ゲート出力を直接アクセスするローバストテスト方式を提案した．さらに，組込み電流テストのための回路設計とテスト生成について提案した．これは単位時間内の電流の積分値を計測し，その値により正常/故障を判定する回路であり，また，この回路に適した動的電流値がほぼ等しいテスト系列の生成を行った．

まずメモリについては，セルアレイの動作に着目し，1ワード線上のすべてのセルの同時テスト可能な組込みテスト方式を提案した．また，この方式に適したテスト手続き，テスト回路をも提案した．対象故障として，その検出が最も困難であると考えられているパターン依存故障，および，デコーダ，読出し・書込み回路の故障を考えた．これらの故障に対し，提案したテスト方式を用いることにより，これまで報告されているテスト系列が $O(N)$ であるのに比べて， $O(N^{1/2})$ のテスト系列でテスト可能であり，より大幅なテスト時

間の減少となることを確認できた。また、パターン依存故障に対し、提案したテスト手続きは $302N^{1/2}$ の準最適なテスト系列を生成する。さらに、本テスト方式に必要なテスト用回路は、シフトレジスタ、カウンタなどを用いて比較的簡単な機構で実現でき、その付加ハードウェア量も記憶容量が増加するにつれ無視できるほど小さくなった。

次に、CMOS 組合せ回路のトランジスタ開放故障に対するテスト容易化設計の提案とテスト生成を行った。開放故障が顕在化されたとき、ゲートが見かけ上トライ・ステート素子になることを利用し、回路外部からゲート出力をアクセスし故障を検出するテスト容易化設計方式の提案を行った。この方式はハザードなどの影響を受けずに開放故障のテストを行えるローバストテストである。本方式は縮退故障にも適用でき、また、順序回路のテストにも拡張できる。さらに、回路内部の再収斂ゲート出力点をテスト点とし、テスト点数の減少をはかった。これによりテストのローバスト性を失うことなくテスト点数を全ゲート数の 13 ~ 60 % に減少させることができ、付加面積オーバヘッドの減少を可能にした。また、テスト生成の実験結果から、生成されたテストはハザードのないローバストテストであることが確認できた。

さらに、組込み電流テストのためのテスト用回路の提案とその回路に適したテスト生成について検討した。CMOS 回路において本質的に生じる動的電流の影響を少なくするために、提案したテスト用回路は、動的電流を含むある単位時間内の電流の積分値を計測し、その値により正常/故障の判定をする。この計測方式により比較的分解能が高く、また、高速の電流テストが行える。SPICE3 シミュレータにより、要求した機能を満足する回路であることを確認した。さらに、テスト用回路の構成は比較的簡単で、かつ、その自己テストも可能である。レイアウト設計を行った結果、回路サイズは $134 \times 89 \mu\text{m}^2$ であった。また、提案した回路に適したテスト生成手法について検討した。これは、テスト系列中のすべてのテストベクトル間での動的電流値をほとんど等しくするために、任意の連続する 2 テストベクトル間でのゲート出力値の変化数がほぼ等しいテスト系列を生成する。このテスト系列は従来の縮退故障のテストのテスト生成プログラムの一部を利用することにより容易に生成でき、また、縮退故障に対するテストベクトル数の 9 ~ 60 % のテストベクトル数となり、大幅にテストベクトル数を減少できた。

最後に今後の課題について述べる。メモリの組込みテストではテスト用回路をチップ内に組込むため、その耐故障性あるいは自己テストについて検討し、組込み自己テストとし

て実現する必要がある。また、開放故障に対するテスト容易化設計については、テスト用周辺回路の詳細な設計、自己テスト性、および、テスト点数を一層減少させるための代表テスト点の決定方法、テストベクトル数の削減などについて検討を要する。さらに電流テストについては、提案したテスト用回路と被テスト回路との総合的な回路シミュレーションを行い、テスト用回路の配置問題など、実用性についての検討が必要である。テスト生成においては、変化数のみを考慮した経路の設定以外に、テスト生成を成功しやすくするための経路の設定方法についても検討を要する。

論理回路は、その機能や構造から組合せ回路、順序回路、メモリ、PLAなどに分類される。本論文では、順序回路についてはほとんど検討をしていないため、電流テストの順序回路への拡張が必要である。また、マイクロプロセッサを代表とする高機能化した回路は、上記した回路が1チップ化した集合体と考えることができるため、本論文で提案したテスト方式を統合化したテスト手法についての検討も必要である。

謝 辞

本研究は大阪大学大学院工学研究科応用物理学専攻において、樹下行三教授の御指導のもとで行ったものである。本研究を遂行するにあたり、終始御指導を賜り、また、有益な議論、および、御助言を頂きました樹下行三教授に心より感謝致します。本論文の作成に関し、詳細な御検討、貴重な御教示を頂きました大阪大学工学部一岡芳樹教授、同大学産業科学研究所岩崎裕教授に深く感謝致します。同じく本論文の作成において、有益な御教示を頂きました同大学工学部増原宏教授、志水隆一教授、南茂夫教授、中島信一教授、興地斐男教授、同大学超高温理工学研究施設後藤誠一教授、同大学産業科学研究所豊田順一教授に深く感謝致します。さらに、終始有益な御助言、御討論を頂きました同大学工学部小松雅治助教授、板崎徳禎博士に深く感謝致します。

秋田大学大学院鉱山学研究科電子工学専攻において、本分野の研究を行う機会、および、メモリのテストについて御指導を賜り、また、有益な議論を頂きました秋田大学鉱山学部成田裕一教授、玉本英夫助教授に心より感謝致します。また、有益な御助言、御討論を頂きました同大学鉱山学部五十嵐隆治講師に深く感謝致します。

樹下研究室の諸氏には一方ならぬ御支援を頂きました。特に、和田恭司氏には、電流テストのテスト生成プログラムの作成に御協力頂きました。ここに記して感謝致します。

参考文献

- (1) M.A. Breuer and A.D. Friedman : "Diagnosis and reliable design of digital systems", Computer Science Press, Rockville, MD (1976).
- (2) 樹下行三, 藤原秀雄 : "デジタル回路の故障診断 (上)", 工学図書 (1983).
- (3) 玉本英夫 : "論理回路の故障診断", 日刊工業新聞社 (1983).
- (4) 樹下行三, 浅田邦博, 唐津修 : "VLSI の設計 II", 岩波書店 (1985).
- (5) H. Fujiwara : "Logic testing and design for testability", MIT Press, Cambridge, Mass. (1985).
- (6) E.J. McCliskey : "Logic design principles, With emphasis on testable semicustom circuits", Prentice-Hall, Englewood Cliffs, NJ (1986).
- (7) V.D. Agrawal and S.C. Seth : "Test generation for VLSI chips", IEEE Computer Society Press, Washington, D.C. (1988).
- (8) M. Abramovici, M.A. Breuer and A.D. Friedman : "Digital systems testing and testable design", Computer Science Press, New York, NY (1990).
- (9) T.W. Williams and K.P. Parker : "Design for testability - A survey", Proc. IEEE, Vol.71, No.1, pp.98-112 (January 1983).
- (10) B. Könemann, J. Mucha and G. Zwiehoff : "Built-in test for complex digital integrated circuits", IEEE J. Solid-State Circuits, Vol.SC-15, No.3, pp.315-319 (June 1980).
- (11) B. Könemann, J. Mucha and G. Zwiehoff : "Built-in logic block observation technique", Proc. Int. Test Conf., pp.37-41 (1979).
- (12) E.J. McCluskey : "Built-in self-test techniques", IEEE Design and Test, Vol.2, No.2, pp.21-28 (April 1985).
- (13) E.J. McCluskey : "Built-in self-test structures", IEEE Design and Test, Vol.2, No.2, pp.29-36 (April 1985).
- (14) R.L. Wadsak : "Fault modeling and logic simulation of CMOS and MOS integrated circuits", Bell Syst. Tech. J., Vol.57, No.5, pp.1449-1474 (May-June 1978).

- (15) J. Galiay, Y. Crouzet and M. Vergniault : "Physical versus logic fault models MOS LSI circuits : Impact on their testability", IEEE Trans. Comput., Vol.C-29, No.6, pp.527-531 (June 1980).
- (16) M.W. Levi : "CMOS is most testable", Proc. Int. Test Conf., pp.217-220 (1981).
- (17) Y.K. Malaiya and S.Y.H. Su : "A new fault model and testing technique for CMOS devices", Proc. Int. Test Conf., pp.25-34 (1982).
- (18) J.M. Acken : "Testing for bridging faults (shorts) in CMOS circuits", Proc. 20th Design Automation Conf., pp.717-718 (1983).
- (19) C.F. Hawkins and J.M. Soden : "Electrical characteristics and testing considerations for gate oxide shorts in CMOS ICs", Proc. Int. Test Conf., pp.544-555 (1985).
- (20) J.P. Shen, W.Maly and F.J. Ferguson : "Inductive fault analysis of MOS integrated circuits", IEEE Design and Test, Vol.2, No.6, pp.12-26 (December 1985).
- (21) F.J. Ferguson and J.P. Shen : "Extraction and simulation of realistic CMOS faults using inductive fault analysis", Proc. Int. Test Conf., pp.475-484 (1988).
- (22) L.K. Horning, J.M. Soden, R.R. Fritzemeier and C.F. Hawkins : "Measurements of quiescent power supply current for CMOS ICs in production testing", Proc. Int. Test Conf., pp.300-309 (1987).
- (23) J.P. Hayes : "Detection of pattern-sensitive faults in random-access memories", IEEE Trans. Comput., Vol.C-24, No.2, pp.150-157 (February 1975).
- (24) J. Knaizuk, Jr. and C.R.P. Hartmann : "An algorithm for testing random access memories", IEEE Trans. Comput., Vol.C-26, No.4, pp.414-416 (April 1977).
- (25) S.M. Thatte and J.A. Abraham : "Testing of semiconductor random access memories", Proc. 7th Int. Symp. Fault-Tolerant Comput., pp.81-87 (1977).
- (26) J. Knaizuk, Jr. and C.R.P. Hartmann : "An optimal algorithm for testing stuck-at faults in random access memories", IEEE Trans. Comput., Vol.C-26, No.11, pp.1141-1144 (November 1977).
- (27) R. Nair, S.M. Thatte and J.A. Abraham : "Efficient algorithms for testing semiconductor random-access memories", IEEE Trans. Comput., Vol.C-27, No.6, pp.572-576 (June 1978).
- (28) D.S. Suk and S.M. Reddy : "An algorithm to detect a class of pattern sensitive faults in

- semiconductor random access memories", Proc. 9th Int. Symp. Fault-Tolerant Comput., pp.219-226 (1979).
- (29) J.P. Hayes : "Testing memories for single-cell pattern-sensitive faults", IEEE Trans. Comput., Vol.C-29, No.3, pp.249-254 (March 1980).
- (30) D.S. Suk and S.M. Reddy : "Test procedures for a class of pattern-sensitive faults in semiconductor random-access memories", IEEE Trans. Comput., Vol.C-29, No.6, pp.419-429 (June 1980).
- (31) D.S. Suk and S.M. Reddy : "A march test for functional faults in semiconductor random access memories", IEEE Trans. Comput., Vol.C-30, No.12, pp.982-985 (December 1981).
- (32) M. Marinescu : "Simple and efficient algorithms for functional RAM testing", Proc. Int. Test Conf., pp.236-239 (1982).
- (33) M.S. Abadir and H.K. Reghbati : "Functional testing of semiconductor random access memories", ACM Comput. Surv., Vol.15, No.3, pp.175-198 (September 1983).
- (34) K.K. Saluja and K. Kinoshita : "Test pattern generation for API faults in RAM", IEEE Trans. Comput., Vol.C-34, No.3, pp.284-287 (March 1985).
- (35) K.K. Saluja and K.T. Le : "Testable design of large random access memories", Integration, VLSI J., Vol.2, No.4, pp.309-330 (December 1984).
- (36) T. Sridhar : "A new parallel test approach for large memories", Proc. Int. Test Conf., pp.462-470 (1985).
- (37) 樹下行三 : "コンパクトテスト法を用いたメモリの組込み検査法", 情処学電子装置設計技術研資, 11-4 (December 1981).
- (38) K. Kinoshita and K.K. Saluja : "Built-in testing of memory chips", Technical Report, EE8335, University of Newcastle, Australia (1983).
- (39) Y. You and J.P. Hayes : "A self-testing dynamic RAM chip", Proc. M.I.T. Conf. Advanced Research in VLSI, pp.159-168 (1984).
- (40) K. Kinoshita and K.K. Saluja : "Built-in testing of memory using on-chip compact testing scheme", Proc. Int. Test Conf., pp.271-281 (1984).
- (41) Z. Sun and L.T. Wang : "Self-testing of embedded RAMs", Proc. Int. Test Conf., pp.148-156 (1984).

- (42) M. Nicolaidis : "An efficient built-in self test scheme for functional test of embedded RAMs", Proc. 15th Int. Symp. Fault-Tolerant Comput., pp.118-123 (1985).
- (43) 玉本英夫, 作左部博朋, 成田裕一 : "アドレス系回路とデータ系回路を分離した IC メモリの組込み検査方式", 信学論(D), Vol.J69-D, No.11, pp.1556-1564 (November 1986).
- (44) 三浦幸也, 玉本英夫, 成田裕一 : "半導体メモリの機能検査に対する組み込み検査", 信学論(D), Vol.J69-D, No.10, pp.1416-1423 (October 1986).
- (45) 三浦幸也, 玉本英夫, 成田裕一 : "1ワード線上の複数個のセルの同時テストを可能にした半導体メモリの組込みテスト", 信学論(D), Vol.J70-D, No.6, pp.1116-1125 (June 1987).
- (46) Y.M. El-zig : "Automatic test generation for stuck-open faults in CMOS VLSI", Proc. 18th Design Automation Conf., pp.347-354 (1981).
- (47) K.W. Chiang and Z.G. Vranesic : "Test generation for MOS complex gate networks", Proc. 12th Int. Symp. Fault-Tolerant Comput., pp.149-157 (1982).
- (48) R. Chandramouli : "On testing stuck-open faults", Proc. 13th Int. Symp. Fault-Tolerant Comput., pp.258-265 (1983).
- (49) S.K. Jain and V.D. Agrawal : "Test generation for MOS circuits using D-algorithm", Proc. 20th Design Automation Conf., pp.64-70 (1983).
- (50) H.K. Lee, D.S. Ha and K. Kim : "Test pattern generation for stuck-open faults using stuck-at test sets in COMS combinational circuits", Proc. 26th Design Automation Conf., pp.345-350 (1989).
- (51) S.M. Reddy, M.K. Reddy and J.G. Kuhl : "On testable design for CMOS logic circuits", Proc. Int. Test Conf., pp.435-445 (1983).
- (52) S.M. Reddy, M.K. Reddy and V.D. Agrawal : "Robust tests for stuck-open faults in combinational circuits", Proc. 14th Int. Symp. Fault-Tolerant Comput., pp.44-49 (1984).
- (53) D.L. Liu and E.J. McCluskey : "Designing CMOS circuits for switch-level testability", IEEE Design and Test, Vol.4, No.4, pp.42-49 (August 1987).
- (54) D.L. Liu and E.J. McCluskey : "CMOS scan-path IC design for stuck-open fault testability", IEEE J. Solid-State Circuits, Vol.SC-22, No.5, pp.880-885 (October 1987).
- (55) R. Rajsuman, A.P. Jayasumana and Y.K. Malaiya : "CMOS stuck-open fault detection using

- single test patterns", Proc. 26th Design Automation Conf., pp.714-717 (1989).
- (56) M. Favalli, P. Olivo, M. Damiani and B. Riccò : "Novel design for testability schemes for CMOS IC's", IEEE J. Solid-State Circuits, Vol.SC-25, No.5, pp.1239-1246 (October 1990).
- (57) Y. Miura and K. Kinoshita : "Testable design for stuck-open faults with robustness", IFIP Workshop on Design & Test of ASICs (1990).
- (58) Y. Miura and K. Kinoshita : "Testable design for stuck-open faults with the robustness", Trans. IEICE, Vol.E73, No.8, pp.1294-1300 (August 1990).
- (59) T. Gheewala : "CrossCheck : A cell based VLSI testability solution", Proc. 26th Design Automation Conf., pp.706-709 (1989).
- (60) W. Maly and P. Nigh : "Built-in current testing - Feasibility study", Proc. Int. Conf. Circuit-Aided Design, pp.340-343 (1988).
- (61) D.B.I. Feltham, P.J. Nigh, L.R. Carley and W. Maly : "Current sensing for built-in testing of CMOS circuits", Proc. Int. Conf. Computer Design, pp.454-457 (1988).
- (62) P. Nigh and W. Maly : "Test generation for current testing", IEEE Design and Test, Vol.7, No.1, pp.26-38 (February 1990).
- (63) R.R. Fritzscheier, J.M. Soden, R.K. Treece and C.F. Hawkins : "Increased CMOS IC stuck-at fault coverage with reduced I_{DDQ} test sets", Proc. Int. Test Conf., pp.427-435 (1990).
- (64) F.J. Ferguson, M. Taylor and T. Larrabee : "Testing for parametric faults in static CMOS circuits", Proc. Int. Test Conf., pp.436-443 (1990).
- (65) W. Mao, R.K. Gulati, D.K. Goel and M.D. Ciletti : "QUIETEST : A quiescent current testing methodology for detecting leakage faults", Proc. Int. Conf. Circuit-Aided Design, pp.280-283 (1990).
- (66) Y. Miura, Y. Wada and K. Kinoshita : "Generation of test sequences for current testing", Proc. 1991 Pacific Rim Int. Symp. Fault Tolerant Systems, pp.200-205 (1991).
- (67) 三浦幸也, 和田恭司, 樹下行三 : "組込み電流テストのためのテスト回路とテスト生成手法", 信学論(DI) (投稿中).
- (68) L.W. Nagel : "SPICE : A computer program to simulate semiconductor circuits", ERL Memo., No.ERL-M520, University of California, Berkeley, CA (May 1975).
- (69) T. Quarles, A.R. Newton, D.O. Pederson and A. Sangiovanni-Vincentelli : "Spice3B.1

- user's guide", ERL Memo., University of California, Berkeley, CA (April 1987).
- (70) M.S. Abadir and H.K. Reghbati : "LSI testing techniques", IEEE Micro, pp.34-51 (February 1983).
- (71) J.F. Poage : "Derivation of optimum tests to detect faults in combinational circuits", Proc. Symp. Mathematical Theory of Automata, pp.483-528 (1963).
- (72) D.B. Armstrong : "On finding a nearly minimum set of fault detection test for combinational logic nets", IEEE Trans. Electronic Comput., Vol.EC-15, No.1, pp.66-73 (February 1966).
- (73) F.F. Sellers, M.Y. Hiao and C.L. Bearson : "Analyzing errors with Boolean difference", IEEE Trans. Comput., Vol.C-17, No.7, pp.676-683 (July 1968).
- (74) L. Goldstein : "Controllability/observability analysis of digital circuits", IEEE Trans. Circuits and Systems, Vol.CAS-26, No.9, pp.658-693 (September 1979).
- (75) M.J.Y. Williams and J.B. Argell : "Enhancing testability of large scale integrated circuits via test points and additional logic", IEEE Trans. Comput., Vol.C-22, No.1, pp.46-60 (January 1973).
- (76) S. Funatsu, N. Wakatsuki and T. Arima : "Test generation systems in Japan", Proc. 12th Design Automation Conf., pp.114-122 (1975).
- (77) E.B. Eichelberger and T.W. Williams : "A logic design structure for LSI testing", Proc. 14th Design Automation Conf., pp.462-468 (1977).
- (78) N. Benowitz, D.F. Calhoun, G.E. Anderson, J.E. Bauer and C.T. Joeckel : "An advanced fault isolation system for digital logic", IEEE Trans. Comput., Vol.C-24, No.5, pp.489-497 (May 1975).
- (79) R.A. Frohwerk : "Signature analysis : A new digital field service method", Hewlett-Packard J., Vol.28, pp.2-8 (May 1977).
- (80) J. Losq : "Referenceless random testing", Proc. 6th Symp. Fault-Tolerant Comput., pp.108- 113 (1976).
- (81) T.W. Williams, W. Daehn, M. Gruetzner and C.W. Starke : "Aliasing errors with primitive and non-primitive polynomials", Proc. Int. Test Conf., pp.637-644 (1987).
- (82) J.P. Roth : "Diagnosis of automata failures : A calculus and a method", IBM J. Res. & Dev., Vol.10, No.4, pp.278-291 (July 1966).

- (83) P. Goel : "An implicit enumeration algorithm to generate tests for combinational logic circuits", IEEE Trans. Comput., Vol.C-30, No.3, pp.215-222 (March 1981).
- (84) J. Inoue, T. Matsumura, M. Tanno and J. Yamada : "Parallel testing technology for VLSI memories", Proc. Int. Test Conf., pp.1066-1071 (1987).
- (85) C. Mead and L. Conway : "Introduction to VLSI systems", Addison-Wesley, Reading, Mass. (1980).
- (86) N. Weste and K. Eshraghian : "Principles of CMOS VLSI design : A systems perspective", Addison-Wesley, Reading, Mass. (1985).
- (87) D.A. Pucknell and K. Eshraghian : "Basic VLSI design : Systems and circuits", 2nd ed., Prentice-Hall, Sydney, Australia (1988).

論文リスト

1. 三浦幸也, 玉本英夫, 成田裕一 : "半導体メモリの機能検査に対する組み込み検査", 電子情報通信学会論文誌(D), Vol.J-69D, No.10, pp.1416-1423 (October 1986).
2. 三浦幸也, 玉本英夫, 成田裕一 : "1ワード線上の複数個のセルの同時テストを可能にした半導体メモリの組み込みテスト", 電子情報通信学会論文誌(D), Vol.J-70D, No.6, pp.1116-1125 (June 1987).
3. Yukiya Miura and Kozo Kinoshita : "Testable design for stuck-open faults with robustness", IFIP Workshop on Design & Test of ASICs (1990).
4. Yukiya Miura and Kozo Kinoshita : "Testable design for stuck-open faults with the robustness", Trans. IEICE, Vol.E73, No.8, pp.1294-1300 (August 1990).
5. Yukiya Miura, Yasushi Wada and Kozo Kinoshita : "Generation of test sequences for current testing", Proc. 1991 Pacific Rim Int. Symp. Fault Tolerant Systems, pp.200-205 (1991).
6. 三浦幸也, 和田恭司, 樹下行三 : "組み込み電流テストのためのテスト回路とテスト生成手法", 電子情報通信学会論文誌(DI) (投稿中).