



Title	CADリンク電子ビームテストシステムによるVLSIの故障追跡に関する研究
Author(s)	三浦, 克介
Citation	大阪大学, 1997, 博士論文
Version Type	VoR
URL	https://doi.org/10.11501/3132600
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

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

The University of Osaka

CAD リンク電子ビームテストシステムによる
VLSI の故障追跡に関する研究

1997 年

三浦克介

CAD リンク電子ビームテスト システムによる
VLSI の故障追跡に関する研究

1997 年

三浦克介

内容梗概

本論文は筆者が大阪大学大学院工学研究科電子工学専攻博士前期課程在学中(平成4年4月～平成6年3月)および大阪大学工学部情報システム工学科(平成6年4月～平成8年5月)、大阪大学大学院工学研究科情報システム工学専攻(平成8年5月～)在職中に行ってきたCAD(Computer Aided Design, 計算機援用設計)リンクEB(Electron Beam, 電子ビーム)テストシステムによるVLSIの故障追跡に関する研究についてまとめたものであり、次の6章から構成されている。

第1章は序論であり、本研究の背景、目的ならびにその工学上の意義、および関連分野における本研究の位置付けについて述べ、また本論文の構成を説明する。

第2章では、CADリンクEBテストシステムについて概説する。システムの概要およびこのシステムを用いた従来の故障追跡法について説明し、このシステムを用いてVLSIの故障追跡を行う際の問題点を挙げる。

第3章では、CADレイアウトからの逐次回路抽出法について述べる。ここで提案する手法は、CADリンクEBテストシステムにおけるレイアウト、スキマティック、ネットリスト等のデータの準備・リンクにかかる手間・時間を削減する為、故障追跡に必要な回路データを、階層構造化されたCADレイアウトデータから逐次的に抽出するものである。まず、効率的に回路抽出を行う為のデータ構造について述べ、このデータ構造を利用して回路抽出を行う手順を述べる。最後に、この手法を加算器のレイアウトデータに適用した例を示し、本手法の有用性を示す。

第4章では、階層的故障追跡アルゴリズムについて述べる。CADレイアウトデータから抽出された回路データには信号の流れの向きに関する情報が記述されていないので、このデータを用いて故障追跡を行うためには、信号の流れの向きを判定する必要がある。また、近年、電圧信号波形の時間的パラメータが設計値から外れることが原因で起こる性能故障が顕著になっており、この故障を解析するためには、トランジスタレベルで故障箇所を指摘することが要求される。この章では、このように信号の流れの向きに関する情報が記述されていない回路データを用いて、階層的に故障追跡し、トランジスタレベルで故障箇所を指摘するアルゴリズムを提案する。次に、このアルゴリズムを商用VLSIの回路データ、およびモデルレイアウトデー

(ii)

タに適用した例を示し、提案したアルゴリズムの有用性を示す。

第5章では、第3、4章の成果をふまえ、EBテストシステム、LSIテストシステム、ワークステーション等を統合して構築した自動故障追跡システムについて述べる。まず、システムのハードウェア構成、ソフトウェア構成について述べ、システムの性能をまとめる。次に、システムを商用のCMOS画像処理プロセッサの故障に対して適用し、構築した自動故障追跡システムがVLSIの故障追跡に有効であることを示す。

第6章では、本研究で得られた成果をまとめ、今後の課題について述べる。

関連発表論文および資料

1. 学会誌

- (1) K. Miura, K. Nakamae and H. Fujioka, "Automatic Tracing of Transistor-Level Performance Faults with CAD-Linked Electron Beam Test System," IEICE Trans. Fundamentals, **E77-A**, 3, pp. 539-545 (1994).
- (2) K. Nakamae, R. Nakagaki, K. Miura and H. Fujioka, "Matching of DUT Interconnection Pattern with CAD Layout in the CAD-Linked Electron Beam Test System," IEICE Trans. Electron., **E77-C**, 4, pp. 567-573 (1994).
- (3) K. Miura, K. Nakamae and H. Fujioka, "Automatic Transistor-Level Performance Fault Tracing by Successive Circuit Extraction from CAD Layout Data for VLSI in the CAD-Linked EB Test System," IEICE Trans. Electron., **E78-C**, 11, pp. 1607-1617 (1995).
- (4) K. Miura, K. Nakamae and H. Fujioka, "Hierarchical Fault Tracing for VLSIs with Bi-Directional Busses from CAD Layout Data in the CAD-Linked EB Test System," IEICE Trans. Electron., **E80-C**, 3, pp. 498-502 (1997).
- (5) K. Miura, K. Nakamae and H. Fujioka, "Hierarchical VLSI Fault Tracing by Successive Circuit Extraction from CAD Layout Data in the CAD-Linked EB Test System," Journal of Electronic Testing: Theory and Applications, **10**, 3, pp. 255-269 (1997).

2. 国際会議

- (1) K. Nakamae, K. Miura and H. Fujioka, "Invited: VLSI Testing with CAD-Linked Electron Beam Test System," Microelectronic Engineering, **31**, 1-4, pp. 319-330

(iv)

(1995).

- (2) K. Miura, K. Nakamae and H. Fujioka, "Hierarchical Fault Tracing for VLSI Sequential Circuits from CAD Layout Data in the CAD-Linked EB Test System," Proc. Asia and South Pacific Design Automation Conference 1997, pp. 329–332 (1997).
- (3) K. Miura, K. Nakata, K. Nakamae and H. Fujioka, "Automatic EB Fault Tracing System by Successive Circuit Extraction from VLSI CAD Layout Data," Proc. 6th Asian Test Symposium, to be published.

3. 研究会

- (1) 三浦克介, 田中裕久, 中前幸治, 藤岡 弘, "CAD リンク電子ビームテストシステムによる VLSI の伝搬遅延故障自動追跡法", 日本学術振興会第 132 委員会 第 121 回研究会 (EB テスティングシンポジウム/1992) 資料, pp. 50–55 (1992).
- (2) 三浦克介, 中前幸治, 藤岡 弘, "CAD レイアウトからの逐次回路抽出による VLSI 自動故障追跡法", 日本学術振興会第 132 委員会 第 125 回研究会 (EB テスティングシンポジウム/1993) 資料, pp. 58–63 (1993).
- (3) 中前幸治, 中垣 亮, 三浦克介, 藤岡 弘, "CAD レイアウトと DUT 配線 SEM 画像との高速・高確度マッチング法", 日本学術振興会第 132 委員会 第 125 回研究会 (EB テスティングシンポジウム/1993) 資料, pp. 64–69 (1993).
- (4) 三浦克介, 中前幸治, 藤岡 弘, "EB テストシステムにおける CAD レイアウトからの逐次回路抽出による VLSI の階層的故障追跡法", 日本学術振興会第 132 委員会 第 128 回研究会 (LSI テスティングシンポジウム/1994) 資料, pp. 71–76 (1994).
- (5) 井口雅保, 三浦克介, 中前幸治, 藤岡 弘, "EB テストシステムによる故障診断のための CAD レイアウトからの回路機能抽出", 日本学術振興会第 132 委員会

- 第 128 回研究会 (LSI テスティングシンポジウム/1994) 資料, pp. 77-82 (1994).
- (6) 三浦克介, 中前幸治, 藤岡 弘, “CAD レイアウトからの逐次回路抽出による VLSI 順序回路の階層的故障追跡法”, 日本学術振興会第 132 委員会 第 132 回研究会 (LSI テスティングシンポジウム/1995) 資料, pp. 84-89 (1995).
- (7) 二口一則, 三浦克介, 中前幸治, 藤岡 弘, “EB テストシステムにおけるレイアウト辞書を用いた VLSI 故障追跡法”, 日本学術振興会第 132 委員会 第 132 回研究会 (LSI テスティングシンポジウム/1995) 資料, pp. 90-95 (1995).
- (8) 藤岡 弘, 中前幸治, 三浦克介, “特別招待論文: CAD リンク EB テストシステムによる VLSI 故障診断の新技法”, 電子情報通信学会技術研究報告 ICD95-187, pp. 21-28 (1995).
- (9) 三浦克介, 中前幸治, 藤岡 弘, “CAD レイアウトからの逐次回路抽出による VLSI 順序回路の階層的故障追跡法 (II)”, LSI テスティングシンポジウム/1996 会議録, pp. 75-80 (1996).
- (10) 松原 豊, 三浦克介, 中前幸治, 藤岡 弘, “EB テストシステムによる VLSI 故障追跡支援のためのレイアウトからのスキマティック自動生成”, LSI テスティングシンポジウム/1996 会議録, pp. 81-86 (1996).

4. 学術講演会

- (1) 三浦克介, 中前幸治, 藤岡 弘, “CAD リンク電子ビームテストシステムによる VLSI 伝搬遅延故障自動追跡法”, 1993 年電子情報通信学会秋季大会, C-422 (1993).
- (2) 三浦克介, 中前幸治, 藤岡 弘, “EB テスタにおける CAD レイアウトからの逐次回路抽出による VLSI 自動故障追跡法”, 1994 年電子情報通信学会春季大会, C-615 (1994).

(vi)

- (3) 井口雅保, 三浦克介, 中前幸治, 藤岡 弘, “EB テストシステムにおける故障診断のための CAD レイアウトからの回路情報抽出”, 1994 年電子情報通信学会秋季大会, C-463 (1994).
- (4) 三浦克介, 中前幸治, 藤岡 弘, “EB テストシステムにおける CAD レイアウトからの逐次回路抽出による VLSI の階層的故障追跡法”, 1995 年電子情報通信学会総合大会, C-569 (1995).
- (5) 二口一則, 三浦克介, 中前幸治, 藤岡 弘, “EB テストシステムにおけるレイアウト辞書を用いた VLSI 故障追跡法”, 1995 年電子情報通信学会エレクトロニクスソサイエティ大会, C-457 (1995).
- (6) 三浦克介, 中前幸治, 藤岡 弘, “EB テストシステムにおける CAD レイアウトからの VLSI 順序回路の自動故障追跡法”, 1996 年電子情報通信学会総合大会, C-528 (1996).
- (7) 松原 豊, 三浦克介, 中前幸治, 藤岡 弘, “EB テストシステム VLSI 故障追跡支援のためのレイアウトからのスキマティック自動生成”, 1996 年電子情報通信学会エレクトロニクスソサイエティ大会, C-481 (1996).
- (8) 三浦克介, 中前幸治, 藤岡 弘, “EB テストシステムにおける CAD レイアウトからの VLSI 順序回路の自動故障追跡法 (II)”, 1997 年電子情報通信学会総合大会, C-12-11 (1997).

目次

第1章 序論	1
第2章 CAD リンク EB テスト システム	7
2.1 まえがき	7
2.2 システムの概要	7
2.2.1 EB テスタ	7
2.2.2 CAD データベース	9
2.2.3 CAD リンク EB テスト システム	14
2.2.4 CAD リンク手順	15
2.3 従来の故障追跡手法	17
2.4 システムの問題点	19
2.4.1 測定不能配線	19
2.4.2 波形測定時間	19
2.4.3 プロービング位置	20
2.4.4 試料ステージ移動誤差	21
2.5 むすび	21
第3章 CAD レイアウト からの逐次回路抽出	23
3.1 まえがき	23
3.2 データ構造	24
3.2.1 四分木データ構造体	25
3.2.2 回路データ構造体	27
3.3 処理手順	28
3.3.1 全体の前処理	28
3.3.2 逐次前処理	28
3.3.3 逐次抽出処理	30

3.4	評価	30
3.5	むすび	32
第4章	階層的故障追跡アルゴリズム	34
4.1	まえがき	34
4.2	回路からグラフへの変換	34
4.3	基本的なアルゴリズム	36
4.3.1	DC パス	36
4.3.2	セル端子の入出力判定	37
4.3.3	DC パスラベルによる上流追跡	38
4.4	測定不能配線への対処	40
4.4.1	測定不能配線での下流追跡	40
4.4.2	下流部分および測定不能配線に存在する故障の扱い	42
4.4.3	FIB・EB 統合化システムによる、測定不能配線の故障箇所絞り込み	43
4.5	順序回路の故障追跡	44
4.5.1	測定位相の制御	44
4.5.2	異なる位相における同一配線のテスト	45
4.6	双方向バス配線での故障追跡	47
4.6.1	入出力端子における信号の流れの向き	48
4.6.2	双方向バス配線での測定位相	48
4.6.3	双方向バス配線での故障追跡法	48
4.7	アルゴリズムの詳細	49
4.8	評価	61
4.8.1	トランジスタレベル故障追跡	61
4.8.2	平坦構造レイアウトからの逐次回路抽出による故障追跡	65
4.8.3	階層構造レイアウトからの逐次回路抽出による故障追跡	69
4.8.4	双方向バス配線を含む順序回路の故障追跡	72
4.8.5	プロービング点数の評価	76

4.9	むすび	79
第5章	自動故障追跡システムの構築	80
5.1	まえがき	80
5.2	ハードウェア構成	80
5.3	ソフトウェア構成	81
5.3.1	故障追跡	82
5.3.2	最適プロービング点選択	82
5.3.3	パターンマッチング	83
5.3.4	波形比較	83
5.3.5	EB テスタの制御	84
5.3.6	LSI テスタの制御	84
5.3.7	各プログラムの関係	84
5.4	システム性能	85
5.5	応用	85
5.6	むすび	91
第6章	結論	93
	謝辞	96
	参考文献	98

第1章 序論

近年、VLSIは、あらゆる種類の電気・電子・情報機器に組み込まれており、半導体集積回路技術は近代社会を支える基盤技術の一つとなっている。この半導体集積回路技術は、大きく分けて、1) 設計技術、2) 製造技術、3) 試験(テスト)技術の三つの要素技術から成り立っている。VLSIの高集積化に伴って、これらの要素技術の中でも、テストの困難性が大幅に増加している。なぜなら、この20年間でVLSI内部の素子数が1000倍以上にも達している^[1]のに対し、チップ外部との情報の出入り口である外部入出力ピンの数は10倍程度にしか増えておらず、内部可制御性、内部可観測性が著しく低下したためである。

VLSIのテスト手法は、外部出力ピンのみをテストする外部間接型テストと、外部出力ピンだけではなくVLSI内部の配線をも直接テストする内部直接型テストの二種類に大別することができる^{[2],[3]}。前者の外部間接型テストは、主に、製造したVLSIが設計仕様を満たしているか否か、即ち良品(正常品)であるか不良品であるかを判定する故障検出を行う為に実施される。テストに用いられる装置としては、LSIテスタが一般的である。これに対し、後者の内部直接型テストは、VLSI試作段階において試作品が設計仕様を満たさなかった場合や、量産段階において歩留まりが低い場合に、故障箇所を特定する故障診断(故障同定)、故障原因を究明する故障解析を行う為に実施される。内部直接型テスト装置には、機械プローブを用いるもの、電子や光ビームを用いるものなど様々な種類のものがあり、その故障の症状に応じて使い分けられているが、この代表例としてEB(Electron Beam, 電子ビーム)テスタ^{[4],[5]}と呼ばれる装置がある。EBテスタは、高真空に保たれた鏡筒内で電磁界により電子を集束してVLSIチップ表面に照射し、その点から放出される二次電子の量あるいはエネルギー分布を測定する事により、VLSIチップ表面の形状・電圧等を測定し、VLSIをテストする装置である。高真空を要するため、1) 装置が大掛かりで高価である、2) DUT(Device Under Test, 被検査素子)の交換に時間を要する、といった欠点があるものの、1) DUTに与える影響が少ない(EBプローブの負荷容量

が小さく、非破壊である)、2) 時間的・空間的電圧分布を高分解能で測定可能、3) 位置合わせが容易である、などの利点を有することから、VLSI の故障診断に必要不可欠な装置として VLSI 産業界で広く利用されている。

EB テスタを用いてテストを行う上で、まず問題となるのは、測定を行おうとする目的の配線が DUT 上のどこにあるのかを、いかにして迅速に探索するかである。IC の時代には、回路規模も小さく、半導体集積回路技術を熟知した小人数の技術者によって設計が行われていたので、設計者であれば容易に配線の位置を知ることができた。しかし、現在においては、回路規模が飛躍的に増大し、設計の自動化・分業化が進んだため、設計者といえども、手作業で目的の配線を見つけることはほとんど不可能である。また、テストが設計者ではなく専門のテスト技術者によって行われる場合も多い。このため、個々の設計に関する知識の無いテスト技術者でも故障診断が可能のように、CAD (Computer Aided Design、計算機援用設計) データベースと EB テスタを結合し、テスト作業を支援する手法・システムが相次いで開発・報告された^{[6]-[15]}。これらのシステムは、CAD リンク EB テストシステムと呼ばれている。CAD リンク EB テストシステムでは、CAD データベース中の回路図データ、ネットリストデータ、レイアウトデータおよび EB テスタが相互にリンクされている。例えば、回路図中である配線を指定すると、その配線の名前、その配線に接続している素子の名前、レイアウト中での配線構成ポリゴン等が表示され、EB テスタの観察視野が、その配線を含む領域に移動される。このため、テスト技術者は容易に目的の配線を見つけることができる^[4]。

EB テスタあるいは CAD リンク EB テストシステムを用いて故障箇所を同定する手法として、故障追跡法がある。故障追跡法とは、故障信号が測定された外部出力ピンから出発し、EB テスタで測定した DUT の測定結果と基準データ (シミュレーション結果または良品測定結果) を比較し、故障信号が測定される方向へ追跡を行う手法である。故障追跡法は、基本的な考え方は非常に単純であるが、適用範囲が広いことから、広く用いられている。この手法は、SEM (Scanning Electron Microscope, 走査電子顕微鏡) 像を比較する手法と電圧信号波形 (以後、単に波形と呼ぶ) を比較する手法の二つに大別することができる^[4]。

SEM 像を比較して故障追跡する方法の代表例に、DFI (Dynamic Fault Imaging, 動的故障像法) 法^{[16]–[18]}と呼ばれる手法がある。DFI 法は、CAD リンク機能は利用せず、DUT のチップ全面の電圧像と正常品の同じ領域の電圧像との差分をとることにより故障信号の追跡を行う。この手法では、画像の取得時間が長くかかり、また画像データの記憶容量が膨大になる。最近、SEM 像の取得法を工夫して一画像あたりの SEM 像取得時間を短縮したり、CAD リンク機能を利用するなどして観察領域を限定し、SEM 像の取得数を削減することにより、追跡時間・記憶容量を低減する手法が種々提案されている^{[19]–[28]}が、後述するように性能故障の追跡が困難であるという問題がある。

波形を比較して故障追跡する方法の代表例にガイドッドプローブ法^{[29]–[36]}がある。ガイドッドプローブ法は、CAD リンク機能を利用して、EB テスタで取得した DUT の波形と基準波形 (シミュレーションによって得られる波形あるいは良品の測定によって得られる波形) を比較することにより故障信号を追跡する。

これらの手法には次のような問題点がある。まず、DFI 法などの SEM 像を比較する手法では、信号伝搬遅延や信号の立ち上がり・立ち下がり時間などの時間的パラメータの僅かに違う状態を識別するのは困難である。これに対し、近年、VLSI の微細化にともなって、VLSI 内部の信号伝搬遅延や信号の立ち上がり・立ち下がり時間などの時間的パラメータが設計値から外れてしまうことが原因で起こる性能故障が顕著になっている^[37]。SEM 像を比較する手法では、このような性能故障の追跡は困難である。

また、波形を比較する手法では、CAD リンク機能の利用が不可欠であるが、CAD リンク機能を利用する為には、LVS (Layout Versus Schematic consistency check, レイアウト対回路図検証) と呼ばれるレイアウト検証用のソフトウェアを用いて、レイアウトデータからトランジスタレベルの回路記述を抽出し、元の回路図やネットリストと対応付けしなければならない^[38]。このとき、チップ全面に対してトランジスタレベルの処理を行うため、この実行に長時間を要する。また、標準セルライブラリ等を利用して設計されている VLSI では、ゲートレベルの回路図やネットリストしか存在せず、トランジスタレベルの回路図やネットリストを得るためには、セルライ

ブラリデータの収集を行う必要があるなど、データの準備作業に手間がかかる。半導体産業界においては VLSI 開発期間の短縮、歩留まりの迅速な向上が強く求められており、故障追跡を短時間で行うことが重要な課題となっている。このため、EB テストシステムの利用に際して、CAD データの準備、リンクにかかる時間の短縮化が求められている。この問題に対し、設計時に回路図やネットリストとレイアウトとの対応関係のデータを出力しておき、このデータを CAD リンクに活用する方法が報告されている^[39]が、対応関係のデータが常時保存されていることは希であり、この手法を適用することができないケースも多い。また、性能故障に関しては、トランジスタレベルの故障箇所指摘を行いたいという要求が強い。しかし、ガイドッドプローブ法では、セルレベルの故障箇所指摘しか行えない。

以上に述べたような状況をふまえ、本研究では、CAD データの準備時間の短縮化、トランジスタレベルでの故障箇所指摘を目的とし、CAD レイアウトからの逐次回路抽出による VLSI 故障追跡法を提案し、この手法を用いた自動故障システムの構築を図る。

本論文の構成を図 1.1 に沿って説明する。

第2章では、CAD リンク EB テストシステムについて概説する。まず、システムの概要について説明し、このシステムを用いた従来の故障追跡法について述べる。次に、このシステムを用いて VLSI 故障追跡を行う際の問題点を挙げる。

第3章では、CAD レイアウトからの逐次回路抽出について述べる。本手法では、CAD リンク EB テストシステムにおけるレイアウト、スキマティック、ネットリスト等のデータの準備・リンクにかかる手間・時間を削減する為、階層構造化された CAD レイアウトデータから逐次的に回路抽出を行う。まず、効率的に回路抽出を行う為のデータ構造およびこのデータ構造を利用して回路抽出を行う手順について述べる。最後に、この手法を加算器レイアウトデータに適用した例を示し、本手法の有用性を示す。

第4章では、階層的故障追跡アルゴリズムについて述べる。CAD レイアウトデータから抽出された回路データには信号の流れの向きに関する情報が記述されていないので、このデータを用いて故障追跡を行うためには、信号の流れの向きを判定す

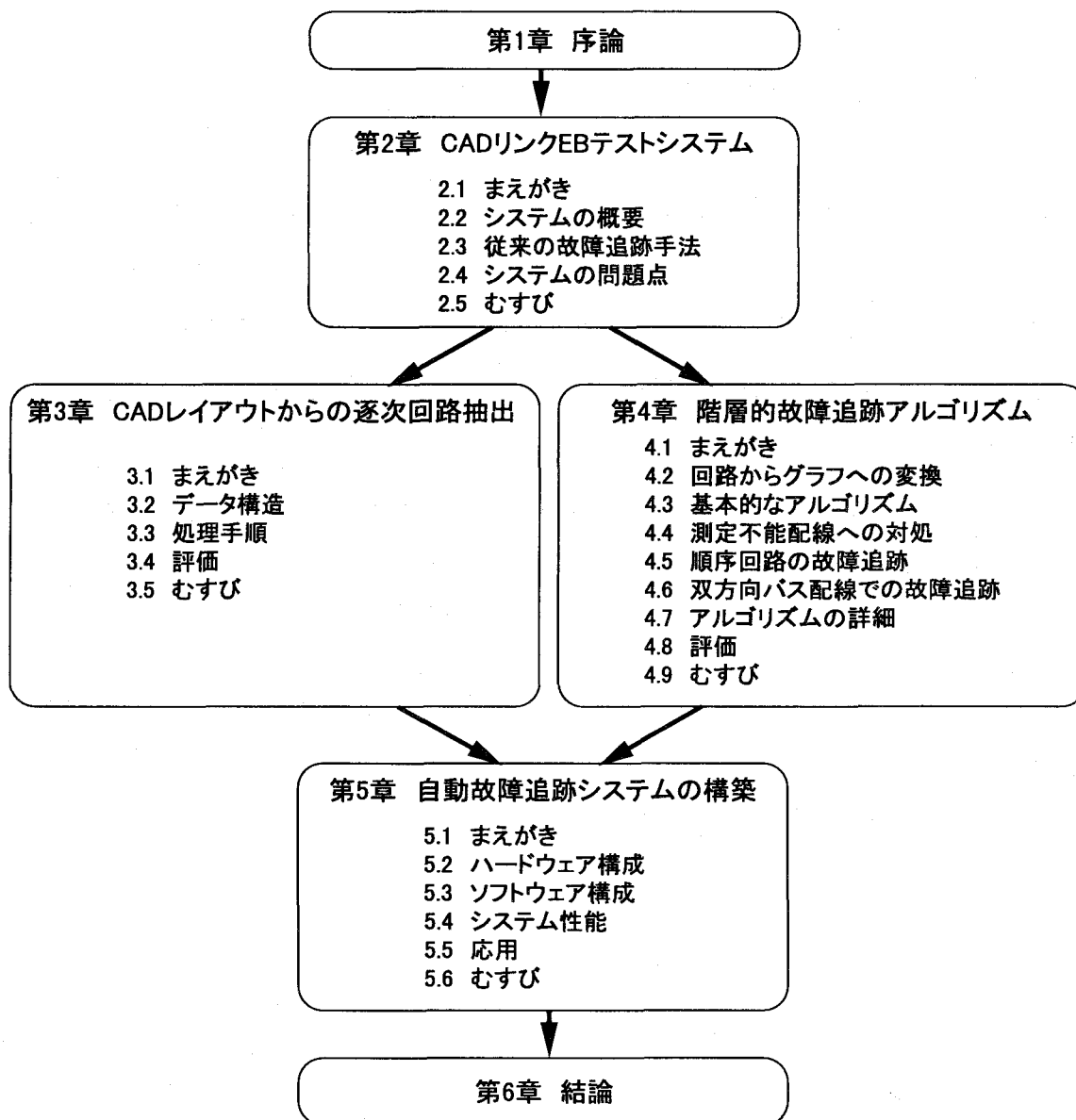


図 1.1 論文の構成

る必要がある。また、性能故障を解析するためには、トランジスタレベルで故障箇所を指摘することが要求される。この章では、このように信号の流れの向きに関する情報が記述されていない回路データを用いて、階層的に故障追跡し、トランジスタレベルで故障箇所を指摘するアルゴリズムを提案する。次に、このアルゴリズムを商用 VLSI の回路データ、およびモデルレイアウトデータに適用した例を示し、本アルゴリズムの有用性を示す。

第 5 章では、第 3、4 章の成果をふまえ、EB テストシステム、LSI テストシステム、ワークステーション等を統合して構築した自動故障追跡システムについて述べる。まず、システムのハードウェア構成、ソフトウェア構成について述べ、システムの性能をまとめる。次に、システムを商用の CMOS 画像処理プロセッサの故障に対して適用し、構築した自動故障追跡システムが VLSI の故障追跡に有効であることを示す。

第 6 章では、本研究で得られた成果をまとめ、今後の課題について述べる。

第2章 CAD リンク EB テスト システム

2.1 まえがき

本章では、CAD リンク EB テストシステムについて概説する。まず、システムの概要およびこのシステムを用いた従来の故障追跡法について説明し、このシステムを用いて VLSI 故障追跡を行う際の問題点を挙げる。

2.2 システムの概要

2.2.1 EB テスタ

EB テスタは、図 2.1 に示すように電子銃、レンズ、走査用コイル、二次電子検出器からなる SEM に、偏向器、絞り、移相器、LSI 駆動用信号発生器を付加して構成される。SEM 部は、通常の SEM に比べて低加速電圧で高空間分解能・高プローブ電流が得られるように工夫されている^{[2], [4]}。

SEM では、配線電圧の高低により、二次電子検出器の信号量が変化し、電圧の低い配線は明るく、高い配線は暗く観察される。また、二次電子検出器として分光器(エネルギーフィルター)を用いることにより、配線上の電圧変化を 10mV 程度の分解能で定量的に測定することができる^[40]。

動作状態の LSI を観察するためには、ストロボ法を用いる^{[2], [4], [40]}。まず、SEM の電子銃から出て来た連続電子ビームを偏向器を通過させて、パルス幅が ns~ps の電子ビームパルスが発生させる。このパルス電子ビームを偏向器と同期して動作する LSI の配線に照射することにより、動作状態の LSI の配線電圧を計測できる。LSI 中の電気信号は正確な周期現象であるから、一周期ごとに必ず同じ電気信号が繰り返し現れる。例えば、図 2.2 上段に示す試料電圧のグラフにおいて、 A_1 と A_2 、 B_1 と

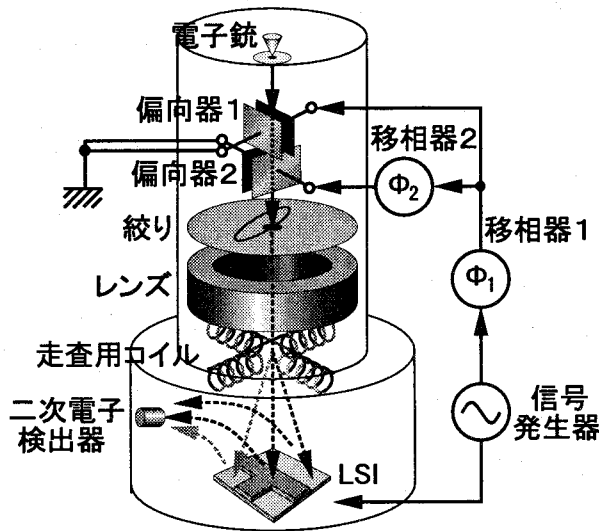


図 2.1 EB テスタの概要

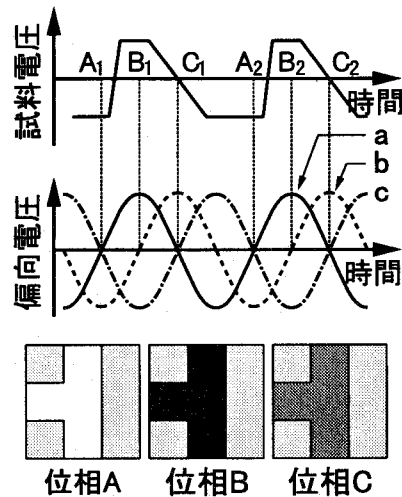


図 2.2 ストロボ法の概要

B_2 、 C_1 と C_2 は、時間が1周期ずれた同位相の時刻で、この時刻における試料電圧はそれぞれ等しい。この時、図 2.2 の中段の偏向器電圧のグラフに示すように、試料電圧に同期した偏向信号 a を偏向器に与えることにより、位相 A (時刻 $t = A_1, A_2, \dots$) の瞬間にだけパルス電子ビームが繰り返し照射される。電子ビームパルスを照射する位相を A に固定して、走査コイルに流す電流を変化させ、電子ビームを二次元的に走査させると、図 2.2 下段の左端に示すように、位相 A における配線の二次元電圧分布 (電圧コントラスト像) を観察することができる。図 2.1 の移相器 1 を操作して偏向信号を図 2.2 中段の b, c に変えて電子ビームパルスを走査させることで、同様に図 2.2 下段中央、右端に示す様に、位相 B, C の電圧コントラスト像がそれぞれ得られる。SEM のこのような観察モードは像モードと呼ばれている。

逆に、走査コイルに流す電流を固定して偏向信号の位相を変化させることで、LSI 上の一点の電圧の時間的变化 (波形) を観察することができる。SEM のこのような観察モードは波形モードと呼ばれている。

2.2.2 CAD データベース

VLSI は、論理設計、レイアウト合成などいくつかの段階を経て設計される。これらの設計段階において、スキマティック、ネットリスト、レイアウトなどの設計データが、人手によって入力され、あるいは自動設計ソフトウェアによって他の設計データから生成される。これらの設計データは CAD データベースに記録される。CAD リンク EB テストシステムではこれらの設計データを、CAD リンク機能のために利用する。本節では、これらの設計データについて説明する。

スキマティック

スキマティックは素子の接続関係を二次元的な図で記述したデータである。素子は、その種類を示す記号で表され、配線は折れ線で表される。素子および配線には、明示的に、あるいは CAD ソフトウェアによって自動的に、固有の名称 (素子名、配線名) が付けられる。

近年の VLSI には 100 万個以上の素子が含まれている。このような大規模な VLSI 全体のスキマティックを一枚の図に表した場合、これを人間が見て理解することは不可能であり、また、これを計算機で処理する場合、長い処理時間を要する。この問題を解決する為、大規模な VLSI のスキマティックは階層構造化されている。階層構造化されたスキマティックでは、配線および素子は、その機能のまとまりに従って区切られ、セルと呼ばれるまとまりにされる。スキマティックにおいてセルは長方形あるいはその機能を表す記号で表され、セル内のスキマティックは別に記録される。セルの外部からは、そのセルの名称 (セル名) と、セル内の配線とセル外の配線の接続点である端子の名称 (端子名)、端子の種類等の情報のみを見ることができる。ここで、端子の種類には入力、出力、入出力、電源、接地などがある。セルの内部回路に注目している時に、そのセルの外部との接続点である端子は、特に外部端子と呼ばれている。

セル内部に多数の配線、素子が含まれる場合は、さらに機能分割が行われ、より小規模なセルが作られる。この機能分割は、セル内部の配線数、素子数が十分少なくなるまで繰り返し行われる。このようにして、チップ全体を最上層とし、十分小

さくされたセルを最下層とする階層構造が作られる。最下層のセルは、インバータ (INV)、NAND、NOR などの基本的な論理素子となる。このように、基本的な論理素子となっている最下層のセルは、とくに論理ゲート、あるいは単にゲートと呼ばれる。これに対して、ゲートよりも上層に位置し、複雑な働きをするセルは、ブロックと呼ばれる。また、最上位のセルはトップセルと呼ばれる。ブロックには、他のセルへの参照が含まれる。他のセルへの参照はセルインスタンスと呼ばれる。セルインスタンスとの区別を明確にするため、参照されるセルの本体は、マスターセルと呼ばれることがある。一つのマスターセルが複数回参照されることがあり、同一のマスターセルを参照する異なるセルインスタンスを区別するため、各セルインスタンスにも固有の名前 (セルインスタンス名) が付けられる。

図 2.3 に階層構造スキマティックの例を示す。このスキマティックは2段の階層構

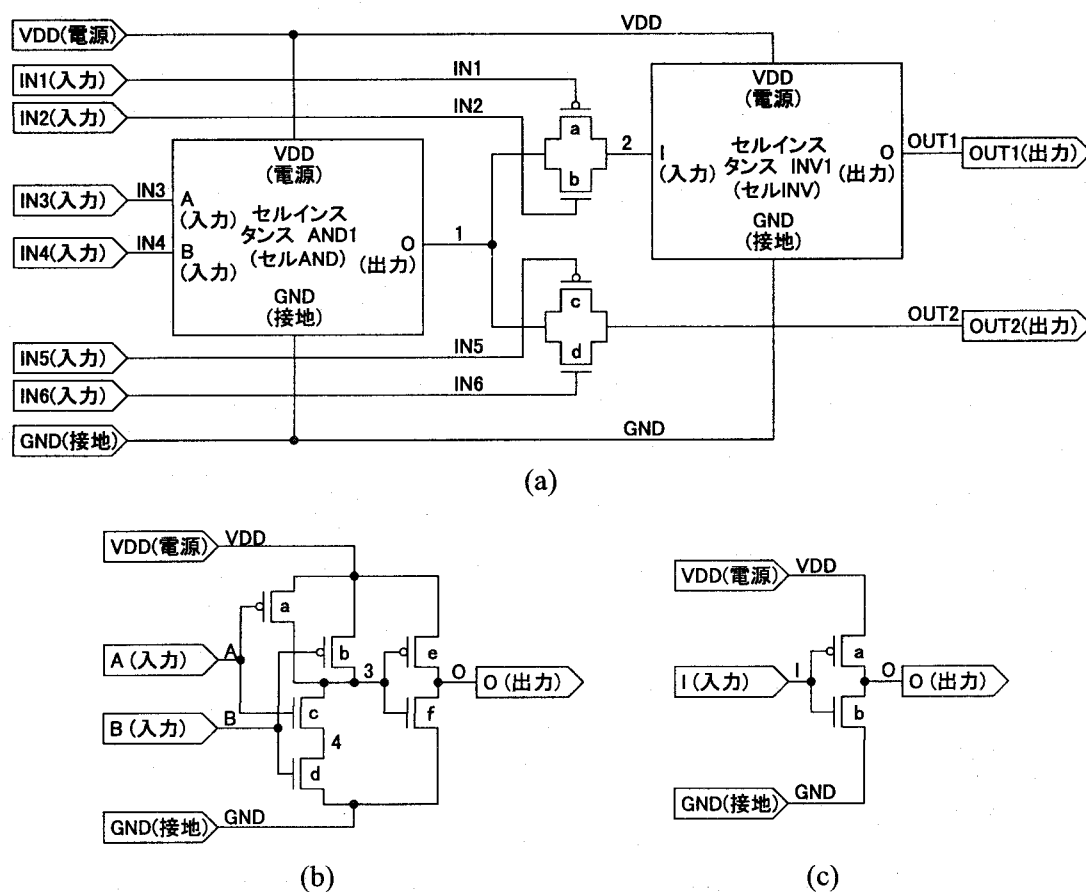


図 2.3 階層構造スキマティックの例、(a) セル TOP、(b) セル AND、(c) セル INV

造をなしている。トップセルは図 2.3 (a) に示されているセル TOP であり、TOP は図 2.3 (b) に示されているセル AND、および図 2.3 (c) に示されているセル INV を参照するセルインスタンスを含んでいる。AND1、INV1 はこれらのセルインスタンス名である。五角形は外部端子を表していて、その中に端子名と端子の種類が書かれている。トップセル TOP の外部端子 (IN1~IN6, OUT1, OUT2, VDD, GND) は、外部入出力パッドに対応する。外部端子の名前とその端子に接続されているセル内の配線の名前は、多くの場合、同じである。図 2.3 の例でも、これらの名前は同じである。1, 2, 3, 4 は外部端子に接続されていない内部配線の名前、a, b, ... は MOS トランジスタの素子名である。セル TOP、セル AND、セル INV で同じ配線名、素子名、端子名が付けられているが、異なるセルで用いられている名前は、別物として区別される。

ネットリスト

ネットリストは、全ての素子の名前とその素子に接続されている配線の名前を列挙したテキスト形式のデータである。ネットリストも、スキマティックと同様に階層構造化されている。ネットリストは、図として表示することができないことを除けば、スキマティックと同等の情報を含んでいる。従って、回路の接続関係の情報が必要となった時、スキマティックとネットリストのどちらを用いても同様な情報を得ることができる。以降において、このようにスキマティックとネットリストを特に区別する必要がない時は、これらをまとめて回路データと呼ぶ。

例として、図 2.3 に示した回路の SPICE (Simulation Program with Integrated Circuit Emphasis, スパイス) 形式のネットリストを表 2.1 に示す。

第 1 行の .SUBCKT で始まる行は、セル TOP の定義の開始を表しており、この行の 2 番目の単語はセル名、残りの単語は外部端子名を示している。第 2, 3 行の X で始まる行は、セルインスタンスを表している。X に続く文字列 (AND1, INV1) はセルインスタンス名である。これらの行の 2 番目以降の単語および数字は、端子に接続されている配線の名前で、最後の単語は参照されるマスターセルの名前を表している。ここで、端子の順番は、マスターセルの .SUBCKT 行での端子の順番と一致

表 2.1 ネットリストの例 (図 2.3 の回路の SPICE 形式ネットリスト)

```

1: .SUBCKT TOP IN1 IN2 IN3 IN4 IN5 IN6 OUT1 OUT2 VDD GND
2: XAND1 IN1 IN2 1 VDD GND AND
3: XINV1 2 OUT1 VDD GND INV
4: Ma 2 IN1 1 VDD PMOS
5: Mb 2 IN2 1 GND NMOS
6: Mc OUT2 IN5 1 VDD PMOS
7: Md OUT2 IN6 1 GND NMOS
8: .ENDS
9:
10: .SUBCKT AND A B O VDD GND
11: Ma 3 A VDD VDD PMOS
12: Mb 3 B VDD VDD PMOS
13: Mc 3 A 4 GND NMOS
14: Md 4 B 3 GND NMOS
15: .ENDS
16:
17: .SUBCKT INV I O VDD GND
18: Ma O I VDD VDD PMOS
19: Mb O I GND GND NMOS
20: .ENDS

```

する。第4～7行のMで始まる行は、MOSトランジスタを示している。Mに続く文字列(a, b, c, d)はMOSトランジスタの名前を示し、2番目以降の単語および数字は、左から順に、それぞれ、ドレイン配線、ゲート配線、ソース配線、基板接続配線、MOSトランジスタの種類を表している。第8行の.ENDSはセル定義の終了を表している。同様に、第10～15行、第17～20行に、それぞれ、セルAND、セルINVの内部回路のネットリストが示されている。

レイアウト

レイアウトデータは、配線、コンタクト、拡散領域、ウェルなどのVLSIの構成要素の形状を記述したデータである。これらの形状は、ポリゴン(多角形)等の基本的な図形要素を組み合わせることにより、表現されている。ポリゴンには、それがどの種類の領域の形状を表しているかを示すため、レイヤと呼ばれる属性が付けられている。レイヤの例としては、ウェル領域を示すウェルレイヤ、トランジスタのソース(S)/ドレイン(D)/ゲート(G)領域を示すSDGレイヤ、ポリシリコン配線領域

を示すポリシリコンレイヤ、金属配線領域を示す金属レイヤ、ソース/ドレイン領域と金属配線およびポリシリコン配線と金属配線を接続するコンタクト領域を示すコンタクトレイヤなどがある。

レイアウトもスキマティックやネットリストと同様に階層構造化される。レイアウトにおいてセルインスタンスは長方形として取り扱われる。この長方形の大きさはマスターセル内の全ての図形要素を囲む最小の外接長方形に等しい。

また、レイアウトでは配線名、素子名、セルインスタンス名、端子名、端子の種類等の情報は、一般的には記述されない。

例として、図 2.3 のスキマティックおよび表 2.1 のネットリストに対応する階層構造レイアウトを図 2.4 に示す。図 2.4 (a)、(b)、(c) はそれぞれセル TOP、セル AND、セル INV のレイアウトを表している。これらのレイアウトにおいて、レイヤは線種、線幅、ハッチングパターンにより表されている。図 2.4 (d) には、レイヤの凡例が示

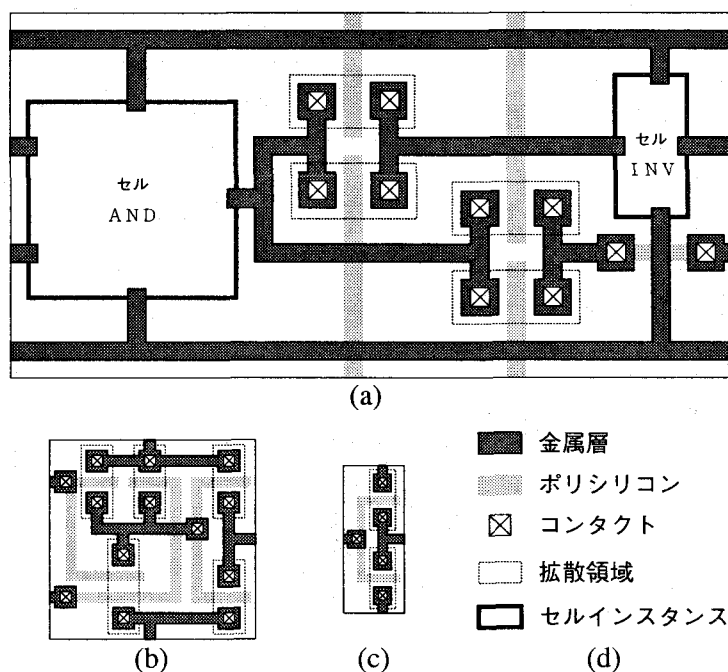


図 2.4 階層構造レイアウトの例 (図 2.3 のスキマティックおよび表 2.1 のネットリストに対応する階層構造レイアウト)、(a) セル TOP、(b) セル AND、(c) セル INV、(d) レイヤの凡例

されている。濃い灰色および薄い灰色で塗りつぶされているポリゴンは、それぞれ金属レイヤ、ポリシリコンレイヤを表し、×印の付けられたポリゴン、点線で囲まれたポリゴンは、それぞれコンタクトレイヤ、SDG レイヤを表している。また、太線で囲まれたポリゴンはセルインスタンスを表している。

図 2.4 (a)、(b)、(c) の各レイアウトにおいて、配線名、素子名は記述されていない。また、(a) のセル TOP のレイアウトには、2つのセルインスタンスが含まれている。これらのセルインスタンスから分かることは、そのマスターセルの名前 (AND おび INV) と、マスターセルの外接長方形の大きさだけである。

2.2.3 CAD リンク EB テストシステム

図 2.5 に、CAD リンク EB テストシステムの概要を示す。システムは EB テスタおよび CAD データベースからなる。CAD データベース中のレイアウト、ネットリ

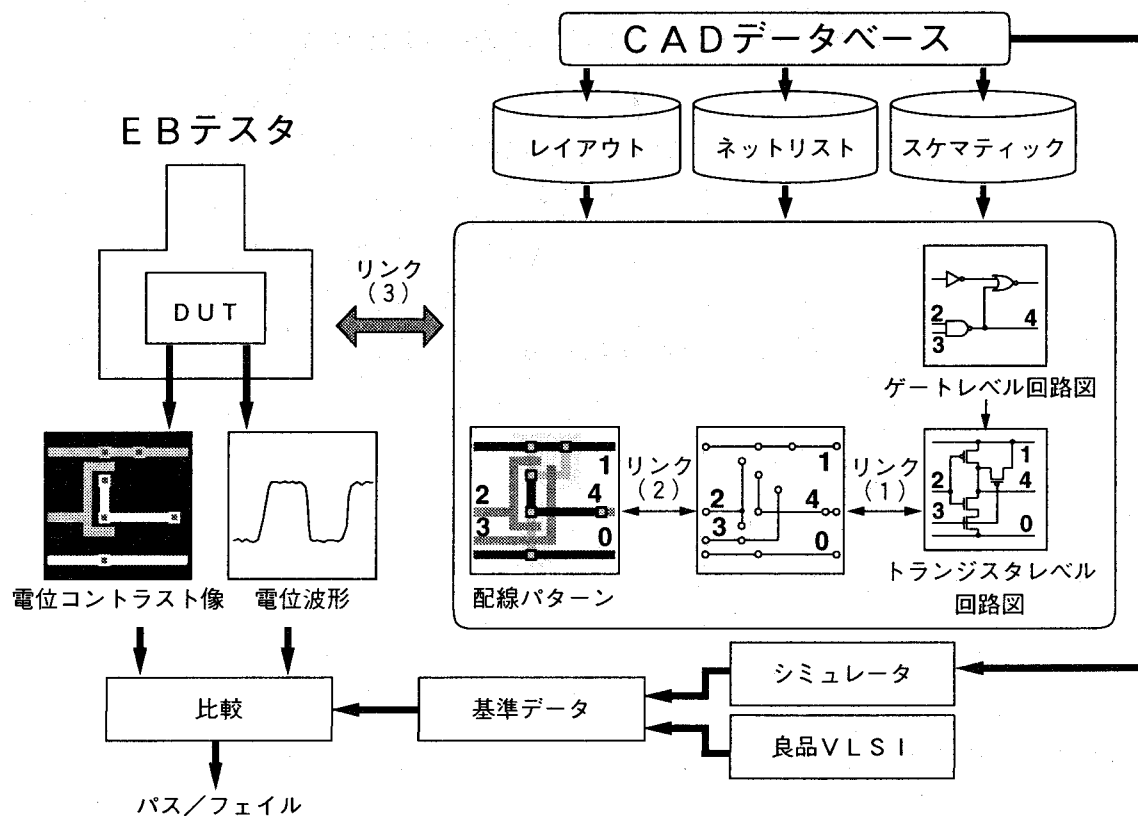


図 2.5 CAD リンク EB テストシステムの概要

スト、スキマティックおよびEB テスタは相互にリンクされている。例えば、スキマティック中のある配線を指定すると、その配線の名前、その配線に接続している素子の名前、レイアウト中での配線構成ポリゴン等が表示され、EB テスタの観察視野が、その配線を含む領域に移動される。このため、テスト技術者は容易に目的の配線を見つけることができる。

測定された電圧コントラスト像や波形は、基準データ (CAD データベース中の設計データをシミュレータにかけることによって得られるシミュレーション結果、あるいは良品 VLSI を測定することによって得られる電圧コントラスト像、波形) と比較することによって正常か故障かの判定がなされ、この判定結果にもとづいて故障診断が行われる。

2.2.4 CAD リンク手順

CAD リンク EB テストシステムにおけるリンクは、図 2.5 に示したように、(1) ネットリストとスキマティックのリンク、(2) レイアウトとネットリストのリンク、(3) EB テスタと CAD データベースのリンク、の 3 つのリンクからなる。それぞれの、リンク手順を以下に示す。

(1) ネットリストとスキマティックのリンク

ネットリストとスキマティックにおいて、対応する配線、素子、セルインスタンスには同じ名前が付けられている。ネットリストとスキマティックそれぞれについて、同じ名前の要素を探索することで、これらのデータがリンクされる。なお、ネットリストはスキマティックから簡単に生成が可能であり、ネットリストが存在せずスキマティックが存在する場合は、スキマティックからネットリストを生成する。

(2) レイアウトとネットリストのリンク

レイアウトとネットリストをリンクする手順の概要を図 2.6 に示す。まずレイアウトとネットリストを LVS と呼ばれるソフトウェアで処理する。LVS は、本来、ネットリストから生成されたレイアウトデータが、元のネットリストと

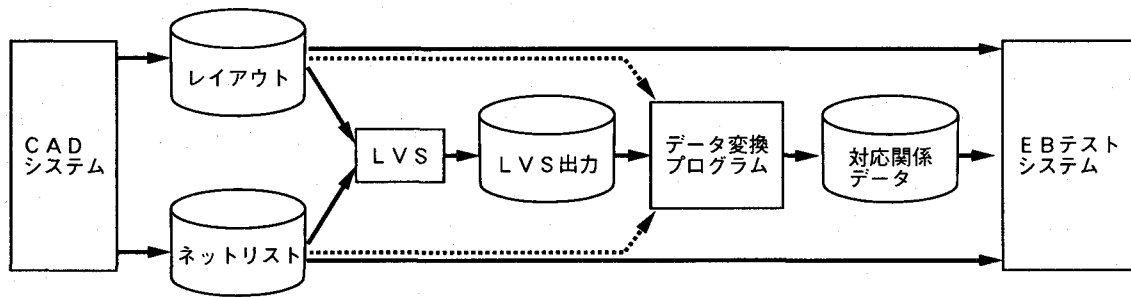


図 2.6 CAD リンクの概要

同一の回路を形成していることを検証する為のソフトウェアで、レイアウトデータからトランジスタレベルのネットリストを抽出して、元のネットリストと比較・照合する。LVS 出力結果には、ネットリスト中の配線、MOS トランジスタとレイアウト中のポリゴンとの対応関係の情報が含まれている。

次に、LVS 出力結果をデータ変換プログラムによって CAD リンク EB テストシステムで処理し易い形式の対応関係データに変換する。この時、図 2.6 に点線の矢印で示したように、元のレイアウトとネットリストを参照する場合もある。CAD リンク EB テストシステムに対応関係データを読み込むことにより、レイアウトとネットリストがリンクされる

(3) EB テスタと CAD データベースのリンク

レイアウト上の何点かの位置座標 (レイアウトデータの座標系における座標) と、それに対応する DUT 上の同じ点の位置座標 (EB テスタの試料ステージの座標系における座標) とから、一方の座標系を他の座標系に変換する式を求める。この時、対応を取る点はオペレータが手動で指定する。この座標変換式によって、レイアウトと EB テスタをリンクする。EB テスタとネットリストをリンクするにはリンク (2), (3) を、EB テスタとスキマティックをリンクするにはリンク (1), (2), (3) をそれぞれ組み合わせることにより行う。

リンク (1), (3) は短時間で行うことができる。しかし、リンク (2) は、全ての配線と MOS トランジスタの比較・照合を行うため、その実行に長時間を要する。また、

標準セルライブラリ等を利用して設計されている VLSI では、ゲートレベルのスキーマティックやネットリストしか存在せず、トランジスタレベルのスキーマティックやネットリストを得るためには、セルライブラリデータの収集を行う必要があるなど、リンク (2) を行うためには、データの準備作業に手間がかかる。この問題を、第 3 章、第 4 章で提案する手法により解決する。

2.3 従来の故障追跡手法

本節では、CAD リンク EB テストシステムを用いた故障追跡の代表的手法であるガイドッドプローブ法について概説し、問題点を指摘する。

ガイドッドプローブ法の概要を図 2.7 に示す。図 2.7 の回路データにおいて、1, 2, ..., 10 は配線名、G1, G2, G3 はゲートのインスタンス名、P は外部出力端子の名前である。今、網かけされて表示されているゲート G1 内に故障があり、ここで発生した故障信号が太線で示した配線を伝搬して外部端子 P に対応する外部出力ピンからチップ外に出力されているとする。ガイドッドプローブ法の手順を以下に示す。

- (1) LSI テスタにより、故障信号が測定される外部出力ピンを特定し、この外部出力ピンに対応する回路データのトップセルの外部端子を p とする。図 2.7 の回

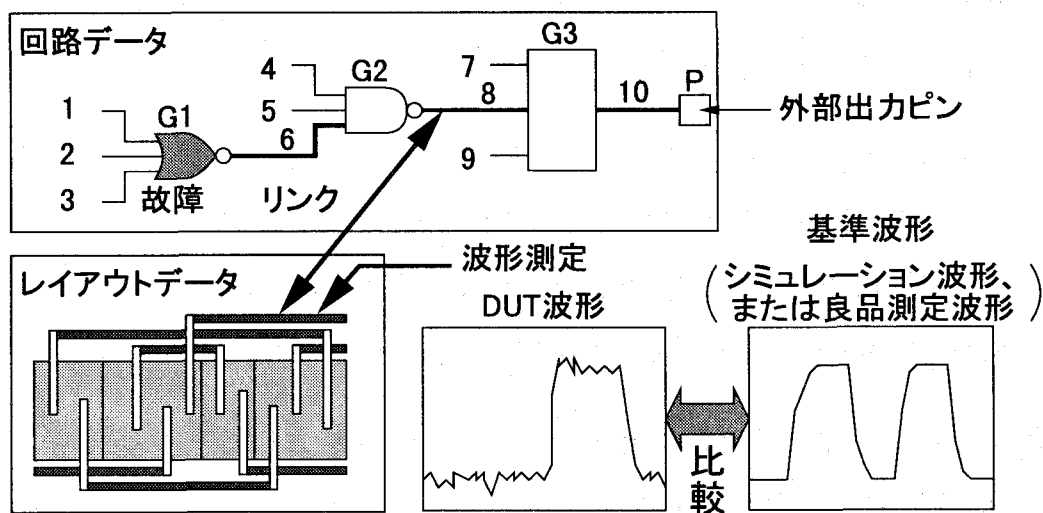


図 2.7 ガイデッドプローブ法の概要

路データの場合、 $p := P$ となる。ここで、 $:=$ は変数への値の代入を意味する。

- (2) 回路データを参照し、 p に接続されている内部配線を現在注目配線 v とする。
 $p = P$ の時、 $v := 10$ となる。
- (3) 回路データを参照し、出力端子が v に接続されているゲートを g とする。 $v = 10$ の時、 $g := G3$ となる。
- (4) 回路データを参照し、 g の入力端子に接続している配線を w_i ($i = 1, 2, 3, \dots$) とする。 $g = G3$ の時、 $w_1 := 7$ 、 $w_2 := 8$ 、 $w_3 := 9$ となる。
- (5) w_i のうち未測定 of 配線 1 本を選択し、図 2.7 で実線の双方向の矢印で示されている CAD リンク機能により、この配線に対応するレイアウトの配線ポリゴンを得る。
- (6) 配線上で EB テスタにより波形測定を行う点を決定し、この座標に EB テスタの試料ステージを移動する。
- (7) EB テスタにより波形の測定を行い、これを DUT 波形とする。
- (8) 図 2.7 で、網かけされた双方向の太い矢印で示されているように、DUT 波形を基準波形 (シミュレーション波形、または良品測定波形) と比較する (基準波形が良品測定波形の場合は、比較を行う前に、DUT と同様に良品の波形測定を行わなければならない)。
- (9) 故障信号が見つかる (DUT 波形と基準波形が異なると判断される) か、配線 w_i の全ての信号が正常 (DUT 波形と基準波形が等しい) と判断されるまで、(5)~(8) の手順を繰り返す。故障信号が見つかった場合は、 $v := w_i$ とする。 $g = G3$ の時、配線 8 (w_2) で故障信号が測定され、 $v := 8$ となる。
- (10) 全ての w_i で正常信号が測定されるまで、(3)~(9) の手順を繰り返す。全ての w_i で正常信号が測定されたとき、ゲート g が故障ゲートである。 $g = G1$ の時、全ての入力配線 1, 2, 3 が正常なので、 $G1$ が故障ゲートであることが同定される。

近年、先に述べたように、VLSI の微細化にともなって、VLSI 内部の信号伝搬遅延や信号の立ち上がり・立ち下がり時間などの時間的パラメータが設計値から外れてしまうことが原因で起こる性能故障が顕著になっている。性能故障では、トランジスタレベルでの故障箇所の指摘が要求されているのに対して、ガイドドプローブ法では、上述したようにゲートレベルでしか故障箇所の指摘が行えない。この問題は、第4章で提案する手法により解決する。

2.4 システムの問題点

第2.2節ではCADリンクに関する問題、第2.3節では従来の故障追跡法の問題について論じたが、本節では、EBテストを用いて波形比較により故障追跡を行う際にシステム上問題となる点を挙げる。

2.4.1 測定不能配線

近年のVLSIでは、配線層の多層化が進んでおり、最低でも2層、多いものでは4層以上の金属配線層を有する。一方、EBテストシステムでは、絶縁保護膜により覆われた下層配線の測定は容量結合法^[41]により行われるが、露出された金属配線を測定する場合に比べて、電圧測定精度が大幅に低下し、正確な波形測定を行うことは困難である^[4]。さらに、下層金属配線が上層の金属配線で覆われている場合には、上層金属によりシールドされ、まったく下層配線の測定を行うことができない。以上のようなことから、このような測定不能な下層配線(以後、測定不能配線と呼ぶ)が多数存在する場合にも故障追跡が可能なアルゴリズムが望まれる。これに関しては、第4.4節で述べる。

2.4.2 波形測定時間

EBテストシステムでは、第2.2.1節で述べたストロボ法により波形の測定を行う。ストロボ法において時間分解能を高くするためには、電子ビームパルス幅(時間軸方向の幅)を短くし、位相の刻み幅を小さくすれば良い。電子ビームパルス幅を短く

すると、S/N 比が悪化するため、平均化回数を増やして、S/N 比を改善する必要がある。従って、時間分解能を高くすると、位相の刻み幅が小さくなり測定を行う位相の点数が増大することと、S/N 比を改善するために平均化回数を増やすことで、波形測定時間が相乗的に長くなる。

しかし、性能故障の認識を行うためには、高い時間分解能での波形測定が必要となる。このため、性能故障の追跡時には、必然的に波形測定時間が長くなってしまふ。特に、順序回路で入力ベクトルシーケンスが非常に長い場合に、その全位相の波形を高い時間分解能で測定するためには、非常に長い時間がかかる。以上のようなことから、できるだけ波形測定時間を短くする手法が望まれる。これに関しては、第 4.5.1 節で述べる。

2.4.3 プロービング位置

EB テスタでの波形測定では、測定を行う点の配線形状や周囲の電磁界の影響により、一次電子および二次電子が影響を受け、測定波形にも好ましくない影響が出ることがある。

このような影響を避けるためには、以下の条件を満たす測定点が望まれる。

- (1) 配線の細い部分では、局所電界効果^[42]により測定される波形の電圧振幅が真の値よりも低くなる場合がある。これを避けるため、なるべく配線の太い部分が望まれる。
- (2) 電子ビーム照射位置誤差に対する許容量を大きくするためには、配線幅の中央部分が望まれる。
- (3) コンタクトホール部は、チップ表面に凹凸があり、測定に悪影響を及ぼす場合がある。これを避けるため、コンタクトホール部以外での測定が望まれる。
- (4) 局所電界効果によるクロストーク^[42]の影響を避けるため、周囲の信号配線との距離が遠い部分が望まれる。
- (5) EB テスタで絶縁膜下の下層金属配線を測定する場合、容量結合法^[41]により

測定を行う。この場合、電圧振幅が低下する、波形のエッジ部がなまる、などの悪影響がある^[4]。このため、できる限り上層の金属配線部分での測定が望まれる。

- (6) 上層金属配線部分に信号が伝搬していないなどの理由で、やむを得ず下層金属配線部分を測定する場合、測定部分を上層金属配線が覆っていると、下層金属配線を測定することができない。下層金属配線を測定する場合は、上層配線によって覆われていない部分を選択する。

以上の条件を満たす測定点を選択する手法が必要である。これに関しては、第5.3.2節で述べる。

2.4.4 試料ステージ移動誤差

EBテストシステムにおいて、観察視野の移動は、X軸、Y軸、それぞれがステップングモーターにより制御される試料ステージを移動させることにより行われる。このステージの移動誤差は、通常、数 μm 程度である。

これに対して、近年のVLSIの金属配線の上層配線幅は $1\mu\text{m}$ 程度以下であり、試料ステージの移動だけでは、正しく目的の配線に電子ビームプローブを照射することはできない。

このため、試料ステージの移動誤差を補整する手法が必要となる。これに関しては、第5.3.3節で述べる。

2.5 むすび

本章では、CADリンクEBテストシステムについて概説した。本システムで故障追跡を行う際の問題点は、以下の通りである。

- (1) CADリンク機能を利用する為には、必要なデータの準備に手間・時間がかかる。
- (2) 近年問題となっている性能故障ではトランジスタレベルの故障箇所指摘が要求されるが、従来の故障追跡法ではセルレベルでしか故障箇所を指摘できない。

- (3) EB テストシステムでは、絶縁膜に覆われた下層配線を測定する場合、電圧測定精度が低下する。
- (4) 順序回路において性能故障の追跡を行う際に全位相の波形を測定すると、非常に時間がかかる。
- (5) 同電圧の配線上でも、プロービング位置によって波形の質が異なる。最適なプロービング位置を求める為には種々の条件を考慮しなければならない。
- (6) 試料ステージの位置決め精度が十分ではなく、プロービング位置の補正作業が必要である。

第3章以降で、以上の問題点を解決する手法を提案し、これをシステムとして実現することを目指す。

第3章 CAD レイアウトからの逐次回路抽出^{[43], [44]}

3.1 まえがき

本章では、CAD レイアウトからの逐次回路抽出について述べる。本手法は、CAD リンク EB テストシステムにおけるレイアウト、スキマティック、ネットリスト等のデータの準備・リンクにかかる手間・時間を削減する為、故障追跡に必要な回路データを、階層構造化された CAD レイアウトデータから逐次的に抽出する。まず、効率的に回路抽出を行う為のデータ構造について説明し、このデータ構造を利用して回路抽出を行う手順を説明する。最後に、この手法をモデル加算器レイアウトデータに適用した例を示し、本手法の有用性を示す。

本章で述べる逐次回路抽出では、以下の条件を満たす階層構造レイアウトデータを対象とする。

- (1) レイアウトは CMOS 技術で設計されており、含まれる素子は MOS トランジスタのみである。
- (2) 階層構造の最下層のセルは、インバータなどの基本的な論理ゲートを構成している。
- (3) セルは互いに重なったり、接したりしない。
- (4) レイアウトに含まれるポリゴンは、水平・垂直なエッジのみを持つ (但し、この条件は、容易に斜めのエッジを持つポリゴンを含むレイアウトに拡張可能である)。

3.2 データ構造

図 3.1 に、故障追跡で用いるデータ構造であるセル構造体を示す。セル構造体は、各セルに対して一つずつ作成される (セルインスタンスに対し一つではなく、何度も参照されるセルに対しても一つだけセル構造体を作成される)。セル構造体は、回路データ構造体と四分木^[45]データ構造体を含んでいる。回路データ構造体は、後の節 (第 4.2 節) で示す回路接続グラフに相当するデータ構造体である。四分木データ構造体は、レイアウトデータを格納するデータ構造体である。これら二つのデータ構造体は、図 3.1 に示されているようにリンク l_1, l_3, l_5 によって相互にリンクされている。回路データ構造体は、トランジスタ構造体、配線構造体、セルインスタンス構造体を含んでおり、これらはリンク l_2, l_4 によって相互にリンクされることにより、回路接続グラフを構成する。また、セルインスタンス構造体はリンク l_6 によって、被参照セルのセル構造体へと片方向にリンクされている。

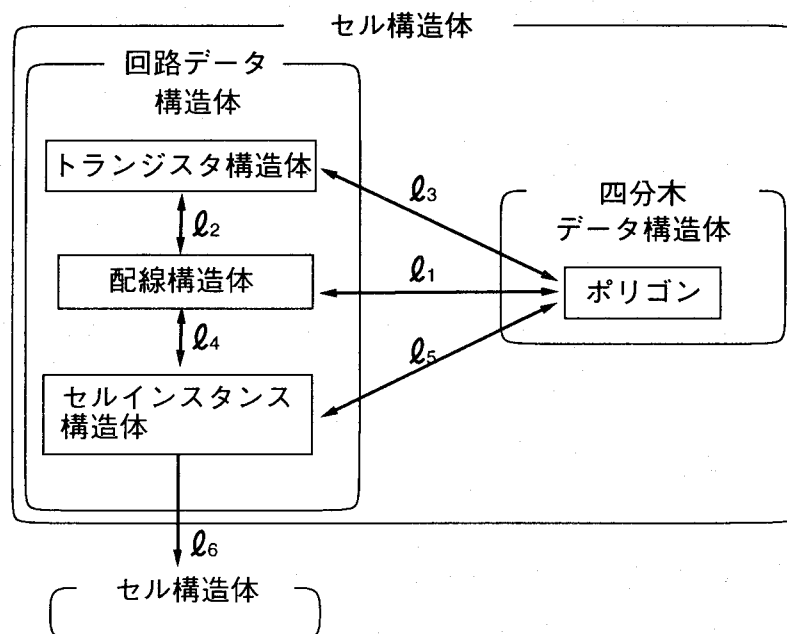


図 3.1 故障追跡で用いられるセル構造体

3.2.1 四分木データ構造体

逐次回路抽出を効率的に行うためには、レイアウトデータ中のポリゴンを高速に検索できなければならない。このため、本手法では、四分木データベース構造を採用している。図 3.2 に示すモデルレイアウトを四分木データ構造に格納した例を図 3.3 に示す。図 3.2 は直列に接続された二つのインバータのレイアウトであり、図の上部に示されているように、線種、線の太さ、ハッチングパターンによりポリゴンのレイヤが表されている。濃い灰色および薄い灰色で塗りつぶされているポリゴンは、それぞれ金属レイヤ、ポリシリコンレイヤを表し、×印の付けられたポリゴンはコンタクトレイヤを表している。細い点線および太い点線で囲まれたポリゴンは、それぞれソース (S)/ドレイン (D) 領域に相当する SD レイヤおよび MOS トランジスタのチャネル領域に相当するチャネルレイヤを表している。また、0, 1, ..., 4 および a, b, ..., r は、それぞれ配線およびポリゴンに付けられたラベルである。

四分木データベース構造は、図 3.3 に示すように、木構造をしており、1つの節点

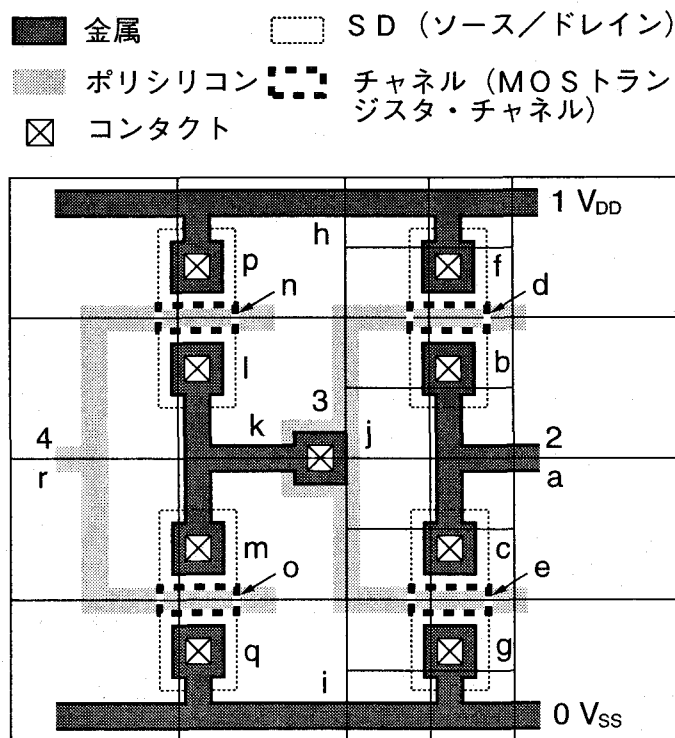


図 3.2 モデル CAD レイアウト

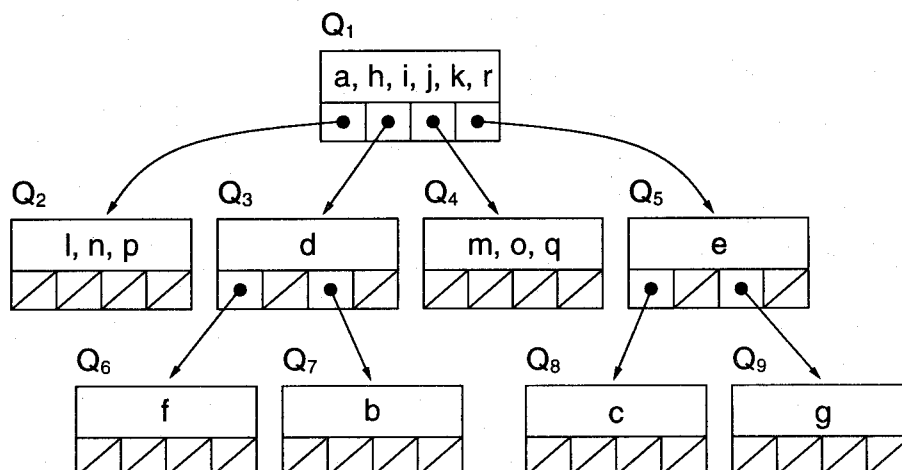


図 3.3 図 3.2 の四分木データ構造

が4つの子木を持つ。木構造の節点は、セルのレイアウト上の長方形の領域に対応している。木構造の最上位の節点がセル全面の外接長方形に対応し、その4つの子節点は、セル全面の外接長方形を縦横に4等分に分割した長方形にそれぞれ対応する。それらの子節点(即ち最上位の節点の孫節点)は、4分割された長方形をさらに4分割した長方形(即ちセル全面の外接長方形を16等分した長方形)に相当する。長方形内に完全に収まるポリゴンがなくなるまで、同様にして、長方形の分割が行われる。

レイアウトの構成要素であるポリゴンのデータは、四分木データ構造の節点に記録される。どの節点に記録するかは、次の様に決められる。

- 四分木の節点 Q に対応する長方形がポリゴン p を完全に囲む長方形の集合の中で最小の大きさのものである時、 p のデータは Q に格納する。

例えば、図 3.2 でポリゴン a, h, i, j, k, r はセル全面を最初に4分割する分割線にかかっており、これらのポリゴンを囲む最小の長方形はセル全面の外接長方形である。従って、図 3.3 の最上位の節点 Q_1 に格納される。また、図 3.2 のポリゴン l, n, p はセルの左上の四半面に完全に収まっており、この四半面をさらに4分割する分割線にかかっているため、これらのポリゴンを囲む最小の長方形は、セルの左上の四半面である。従って、この長方形に対応する図 3.3 の節点 Q_2 に、これらのポリゴンの

データが格納される。

四分木データ構造体は、配線と MOS トランジスタの接続や配線とセルインスタンスの接続を抽出する為にポリゴン同士の接触や重なりを高速に調べるのに適している。例えば、図 3.2 の配線 2 にソース/ドレイン端子が接続している MOS トランジスタを検索する場合を考える。このためには、SD レイヤのポリゴンであるポリゴン b, c に接触しているチャネルレイヤのポリゴンを見付ければ良い。このうち、ポリゴン b について考える。ポリゴン b は図 3.3 の節点 Q_7 に格納されている。この場合、検索は節点 Q_7 とその上位節点 (節点 Q_1, Q_3) について行えば良い。セルと配線の接続についても、同様に、金属レイヤのポリゴンとセルインスタンスレイヤのポリゴンの接触・重なりを調べることによって抽出される。ここで、セルインスタンスレイヤとは、セルインスタンスの外接長方形に相当するポリゴンのレイヤである。

3.2.2 回路データ構造体

回路データ構造体は、トランジスタ構造体、配線構造体、セルインスタンス構造体およびそれらの相互リンク (図 3.1 のリンク l_2, l_4) からなっている。

配線構造体はレイアウトデータ中の導電レイヤ (金属、ポリシコン、SD レイヤ) のポリゴンとコンタクトレイヤのポリゴンとの接触をチェックすることで抽出される。このチェックは、導電レイヤとコンタクトレイヤの AND 操作 (共通部分の抽出) によって行われる。図 3.2 において、金属レイヤであるポリゴン a と SD レイヤであるポリゴン b, c とコンタクトレイヤのポリゴンの AND をとり、共通部分が存在することから、ポリゴン a, b, c が電氣的に接続していることが分かる。これにより、配線 2 の配線構造体が作られ、各ポリゴン構造体と配線構造体の間にヘリンク (図 3.1 のリンク l_1) が張られる。

トランジスタ構造体はチャネルレイヤのポリゴン 1 つに対して 1 つ生成される。1 対 1 対応なので、ポリゴン名をそのまま素子名とする。例えば、図 3.2 のポリゴン d に対してトランジスタ構造体を生成し、素子名を d として、これらの間にリンク (図 3.1 のリンク l_3) を張る。

セルインスタンス構造体はセルインスタンス 1 つに対して 1 つ生成される。マス

ターセルのレイアウトの外接長方形に対応するセルインスタンスレイヤのポリゴン構造体も生成され、セルインスタンス構造体との間にリンク (図 3.1 のリンク l_5) が張られる。これらの構造体の生成、リンク作成は第 3.3.2 節で述べる逐次前処理において行われる。

配線構造体とトランジスタ構造体のリンク l_2 は次のように抽出される。MOS トランジスタにはソース、ゲート、ドレインの3つの端子がある。ソース/ドレイン端子と配線の接続は、チャンネルレイヤのポリゴンに接触している SD レイヤのポリゴンを検索することで行える。ゲート配線の接続は、チャンネルレイヤのポリゴンに重なっているポリシリコンレイヤのポリゴンを検索することで行える。

3.3 処理手順

回路抽出は、全体の前処理、逐次前処理、逐次処理の3段階で行われる。全体の前処理は、最初に1度だけ実行される。逐次前処理は、セルの中身の情報が必要になった時点で、各セルに対して1度だけ実行される。逐次処理は、故障追跡処理と並行して、故障追跡処理の要求に従って実行される。

3.3.1 全体の前処理

全体の前処理では各セルに対してセル構造体を生成する。このとき、回路データ構造体、四分木構造体は生成せず、セル構造体の中身は空にしておく。これらのセルの中身は、セル内の回路情報が必要となった時点でデータが読み込まれ、各構造体が生成される。

3.3.2 逐次前処理

図 3.1 の配線構造体、セルインスタンス構造体、および四分木データ構造体とリンク l_1, l_5, l_6 は、逐次前処理で生成される

逐次前処理は、次の手順で行われる。

(1) ポリゴンマージ

入力 CAD レイアウトデータに含まれるポリゴンデータは、必ずしも必要最小限のものではない。多くの場合、物理的には一つのポリゴンが、データ上では複数のポリゴンに分割されている。これらのポリゴンをエッジベースレイアウト演算法^[46]により一つのポリゴンに結合する処理を行う。

(2) チャネルレイヤと SD レイヤの生成

チャネルレイヤは、ポリシリコンレイヤと SDG レイヤの AND 操作 (共通部分の抽出) により生成する。SD レイヤは、SDG レイヤからポリシリコンレイヤを差し引くことにより生成する。

(3) 四分木データベースの構築

各ポリゴンの位置を調べ、対応する四分木の節点にポリゴン情報を格納する。

(4) セルインスタンスレイヤとセルインスタンス構造体の生成

各セルインスタンスに対して、対応する外接長方形を調べ、この長方形に相当するセルインスタンスレイヤのポリゴンを生成し、対応するセルインスタンス構造体を生成する。このポリゴンとセルインスタンスの間にリンク (図 3.1 のリンク l_5) を張り、同時にセルインスタンス構造体から被参照セルのセル構造体へリンクを張る (図 3.1 のリンク l_6)。

(5) 配線の抽出と配線構造体の生成

導電レイヤ (金属レイヤ、ポリシリコンレイヤ、SD レイヤ) とコンタクトレイヤの接続をチェックし、配線を抽出する。配線構造体を生成し、各配線に固有の配線名を付け、配線を構成するポリゴンとの間に双方向のリンク (図 3.1 のリンク l_1) を張る。

(6) 電源配線の設定

上位のセルにおける電源配線とセル内の配線の接続チェックを行い、電源配線を設定する。トップセルの場合は、オペレータが電源配線を指定するか、レイアウトデータに記入されたテキストラベルにより電源配線を指定する。

3.3.3 逐次抽出処理

以下の処理は、故障追跡アルゴリズムの要求に応じて、逐次実行される。

- (1) MOS トランジスタと配線の接続をチェックする。
- (2) トランジスタ構造体を生成する。
- (3) トランジスタ構造体とその MOS トランジスタのチャネル領域に相当するチャネルレイヤのポリゴンのポリゴン構造体との間にリンクを張る (図 3.1 のリンク ℓ_3)
- (4) 配線構造体とトランジスタ構造体の間にリンクを張る (図 3.1 のリンク ℓ_2)。
- (5) セルインスタンスと配線の接続をチェックする。
- (6) セル内部の配線と上位セルの配線の接続をチェックする。
- (7) セルインスタンス構造体と配線構造体の間にリンクを張る (図 3.1 のリンク ℓ_4)。

これらの処理により、故障追跡を行う部分の回路データ構造が構築され、故障追跡に必要な回路データが得られる。

3.4 評価

レイアウトからの逐次回路抽出手法の有効性を示す為、UNIX ワークステーション上において C 言語を用いて本手法を実装し、これを図 3.4 に示す 16 ビット全加算器の階層構造レイアウトに対して適用した。このレイアウトは、2 層金属 CMOS プロセスで設計されており、約 600 トランジスタを含んでいる。基板コンタクトなどの故障追跡に直接関係のない要素は省略されている。出力配線の一つを指定し、ここから順次回路抽出を行った結果を図 3.5 ~ 3.7 に示す。図 3.5 はトップセルの回路図であり、セル 4BIT-ADD を参照する 4 つのセルインスタンスが含まれている。図 3.6 はセル 4BIT-ADD の回路図であり、セル 1BIT-ADD を参照する 4 つのセルインスタンスが含まれている。図 3.7 はセル 1BIT-ADD の回路図であり、セル 2-1-OA、

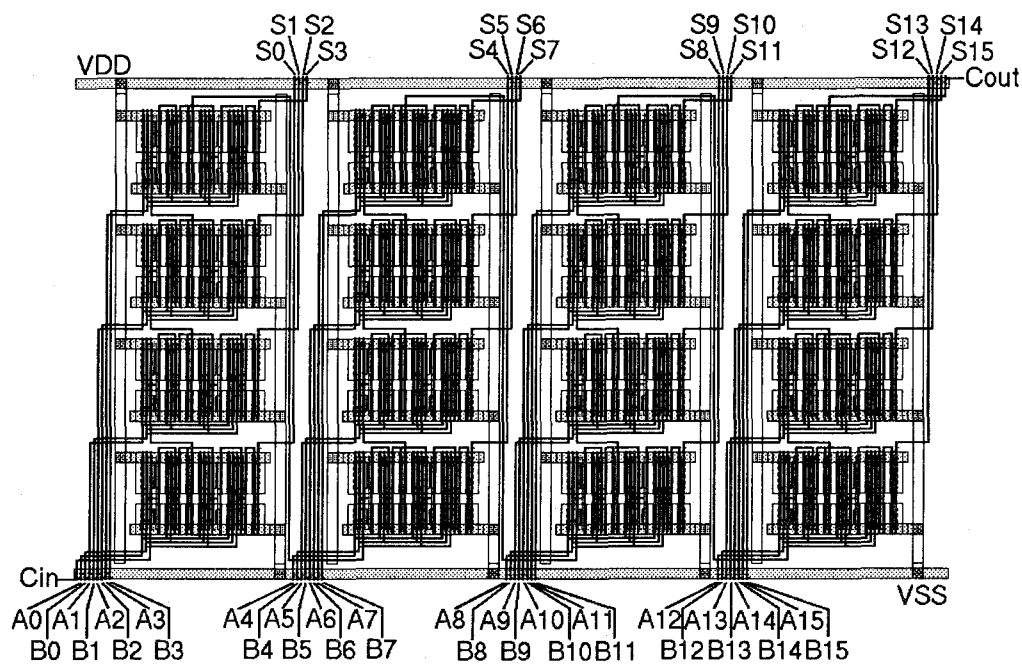


図 3.4 階層構造 16 ビット全加算器のレイアウト

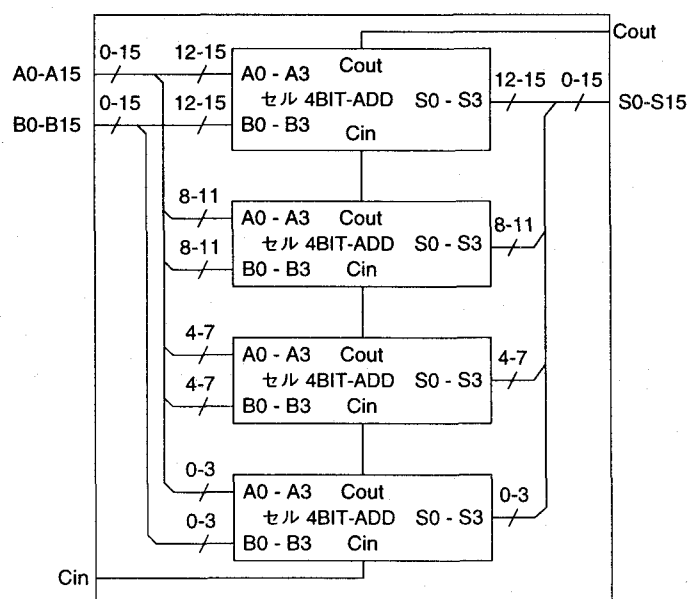


図 3.5 図 3.4 のレイアウトから抽出されたトップレベルの回路図

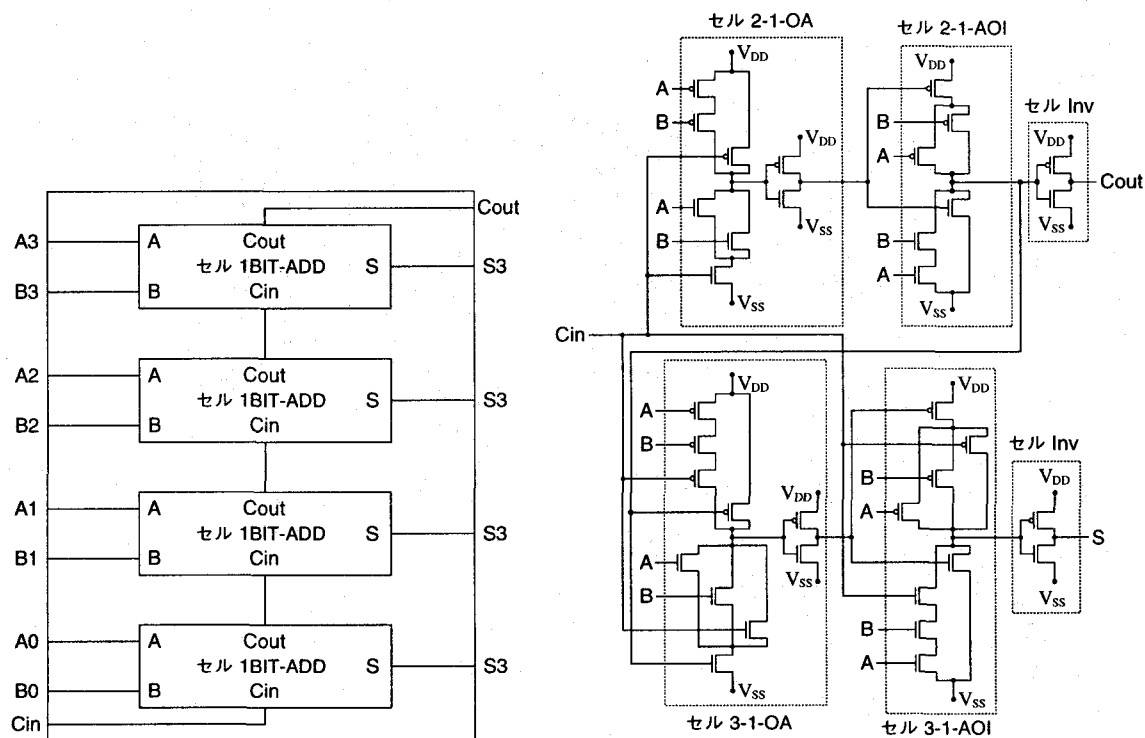


図 3.6 図 3.4 のレイアウトから抽出された第 2 レベルの 4 ビット加算器 (セル 4BIT-ADD) の回路図。

図 3.7 図 3.4 のレイアウトから抽出された第 3 レベルの 1 ビット加算器 (セル 1BIT-ADD) の回路図。

セル 2-1-AOI、セル 3-1-OA、セル 3-1-AOI、セル INV を参照するセルインスタンスが計 6 つ含まれている。本来、これらのセル内の回路図は別に記述されるものだが、ここでは、これらのセルを全て展開して表示している。

以上より、レイアウトデータに対応する階層構造回路データが正しく抽出されていることが確認できた。

3.5 むすび

本章では、故障追跡の為に必要となる階層構造回路データを階層構造レイアウトデータから逐次的に抽出する手法について述べた。本手法の特徴は以下の通りで

ある。

- (1) 四分木データベースにレイアウトデータを格納し、高速に逐次的な回路抽出を行う。
- (2) 回路抽出処理を全体の前処理、逐次前処理、逐次抽出処理の3段階に分け、逐次前処理、逐次抽出処理は故障追跡アルゴリズムの要求に応じて実行する。故障追跡に関係の無い部分は抽出処理が行われないので、全回路抽出に比べ格段に処理量が少ない。

第 4 章 階層的故障追跡アルゴリズム^{[44],[47]–[49]}

4.1 まえがき

本章では、階層的故障追跡アルゴリズムについて述べる。CAD レイアウトデータから回路抽出を行って、これを故障追跡に利用する場合、レイアウトデータには信号の流れの向きに関する情報が記述されていないことが問題となる。また、性能故障を追跡する場合、セルレベルではなく、トランジスタレベルで対象回路を取り扱い、トランジスタレベルで故障箇所を指摘する必要がある。この章では、このように信号の流れの向きに関する情報が記述されていない回路データを用いて、階層的に故障追跡し、トランジスタレベルで故障箇所を指摘するアルゴリズムを提案する。さらに、測定点が下層配線上にある場合の対処法、順序回路の故障追跡法、双方向バス配線に対する対処法についても述べる。最後に、このアルゴリズムを商用 VLSI の回路データおよびモデルレイアウトデータに適用した例を示し、本アルゴリズムの有用性を示す。

4.2 回路からグラフへの変換

本アルゴリズムでは以下の条件を満たす回路を取り扱う。

- (1) 故障追跡の対象となる VLSI は、CMOS デジタル回路である。
- (2) 階層構造レイアウトデータから抽出され、信号の流れの向きに関する情報が記述されていない階層構造回路データを取り扱う。
- (3) 階層構造の最下層では、回路はトランジスタレベルで記述されている。
- (4) 順序回路、双方向バス回路を含んでいても良い。

以降の節で説明する故障追跡アルゴリズムで用いるため、回路をグラフ化する。このグラフを回路接続グラフと呼ぶ。回路接続グラフは、第 3.2.2 節で示した回路

データ構造と対応する。

グラフ化の規則は以下の通りある。

- (1) MOS トランジスタ、セルインスタンス、配線を節点に変換する。
- (2) 端子 (MOS トランジスタのゲート端子、ソース/ドレイン端子、およびセルインスタンスの入力端子、出力端子、入出力端子、電源端子および接地端子) を枝に変換する。

例として、図 4.1 に示す回路をグラフに変換したものを、図 4.2 に示す。図 4.1 は 2 段の階層構造を持つ回路データで、図 4.1 (a) にトップセル TOP、(b) にセル AND、(c) にセル INV の回路データがそれぞれ示されている。これらの回路データにおいて、1, 2, ..., 22 は配線名。a, b, ..., n は MOS トランジスタ名を表している。図 4.2 (a), (b), (c) は、それぞれトップセル TOP、セル AND、セル INV の回路接続グラフである。これらの図中で、“gate”、“s/d” と書かれた枝は、それぞれ MOS トランジスタのゲート端子、ソース/ドレイン端子を意味する。“in”、“out” と書かれた枝は、それぞれセルの入力端子、出力端子を意味する。回路接続グラフは各セルに対して作成され、セルインスタンスに相当する節点には、対応するセルの回路接続グラフを指すポインタを持たせる。

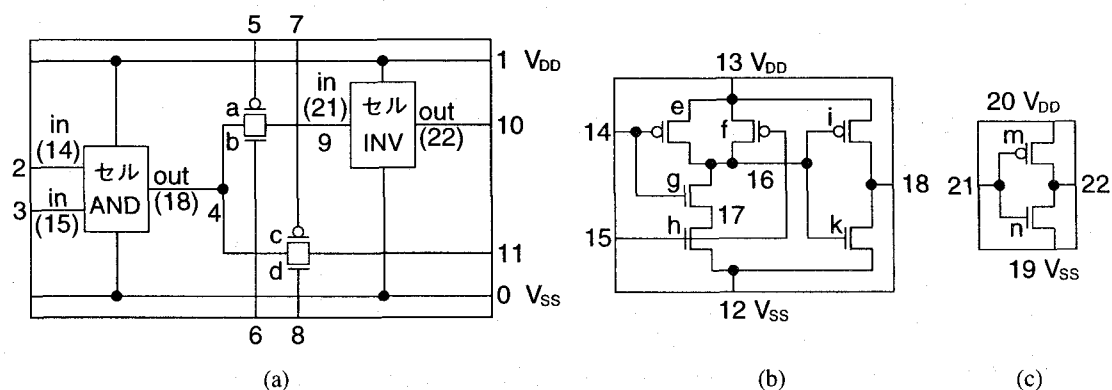


図 4.1 (a) トップセル TOP、(b) セル AND、(c) セル INV の回路データ。1, 2, ..., 22 は配線名。a, b, ..., n は MOS トランジスタ名。

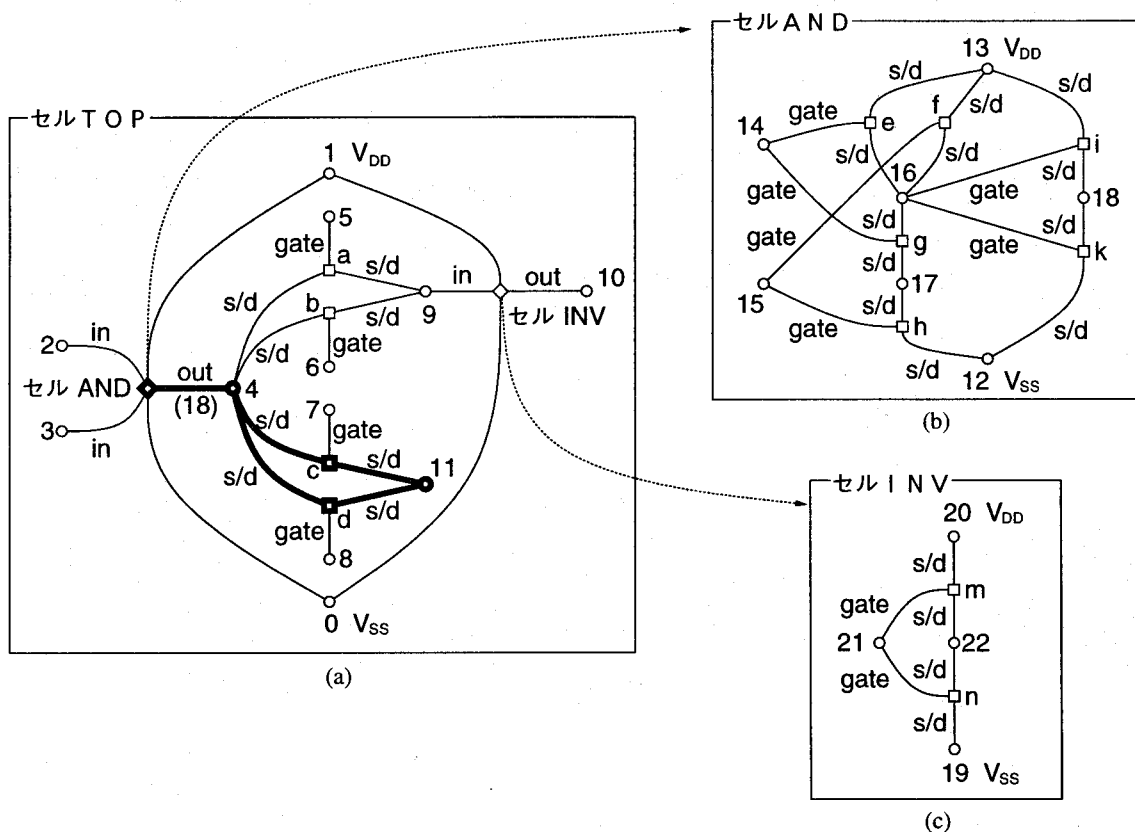


図 4.2 図 4.1 の回路データから変換した、(a) トップセル TOP、(b) セル AND、(c) セル INV の回路接続グラフ。1, 2, ..., 22 のラベルが付けられた○印は配線、a, b, ..., n のラベルが付けられた□印は MOS トランジスタ、◇印はセルインスタンス、gate, s/d, in, out とラベルの付けられた枝は、それぞれゲート端子、ソース/ドレイン端子、入力端子、出力端子。太線は、配線 11 からの DC パス。

4.3 基本的なアルゴリズム

4.3.1 DC パス

以降の節で必要となる DC (Direct Current, 直流) パスを定義する。

定義 4.1 : DC パス

ある配線から、MOS トランジスタのソース/ドレイン端子を経由して電源配線 (V_{DD} , V_{SS})、またはセルインスタンスの出力端子に到達する経路を DC パスと呼ぶ。

例えば、図 4.2 のセル AND において、配線 16 からの DC パスは、 $\{16, e, 13\}$, $\{16, f, 13\}$, $\{16, g, 17, h, 12\}$ である。また、同じく図 4.2 のセル TOP において、配線 11 からの DC パスは、 $\{11, c, 4, B\}$, $\{11, d, 4, B\}$ である。

4.3.2 セル端子の入出力判定

通常、レイアウトデータには信号の流れの向きに関する情報は記述されていない。従って、セルの端子が、入力であるか、出力であるか、入出力であるかを判定する必要がある。

端子の種類の判別は二段階の手続きで行う。第一段階では、注目するセルの端子からセル内部に向かって、以下の経路を探索する。

- (1) MOS トランジスタのソース/ドレイン間を経由して、注目するセル端子と MOS トランジスタのゲート端子とを結ぶ経路。
- (2) 注目するセル端子と V_{DD} , V_{SS} 等の電源配線とを結ぶ DC パス。

これら経路の有無によって、以下の端子を 3 種類に分類する。

タイプ 1：上記 (1) の経路のみが存在する場合。この場合は、セル端子は入力端子である。

タイプ 2：上記 (2) の DC パスのみが存在する場合。この場合は、セル端子は出力端子である。

タイプ 3：上記 (1)、(2) の両方の経路が存在する場合。この場合は、セル端子は入出力端子である可能性と出力端子である可能性がある。

注目するセル内部にさらにセルインスタンスが存在する場合には、そのセルインスタンス内部へも、上述の経路探索を行う。

端子の分類の例を図 4.3 (a) に示す。端子 a, b, c はタイプ 1 (入力)、端子 d はタイプ 2 (出力)、端子 e はタイプ 3 である。

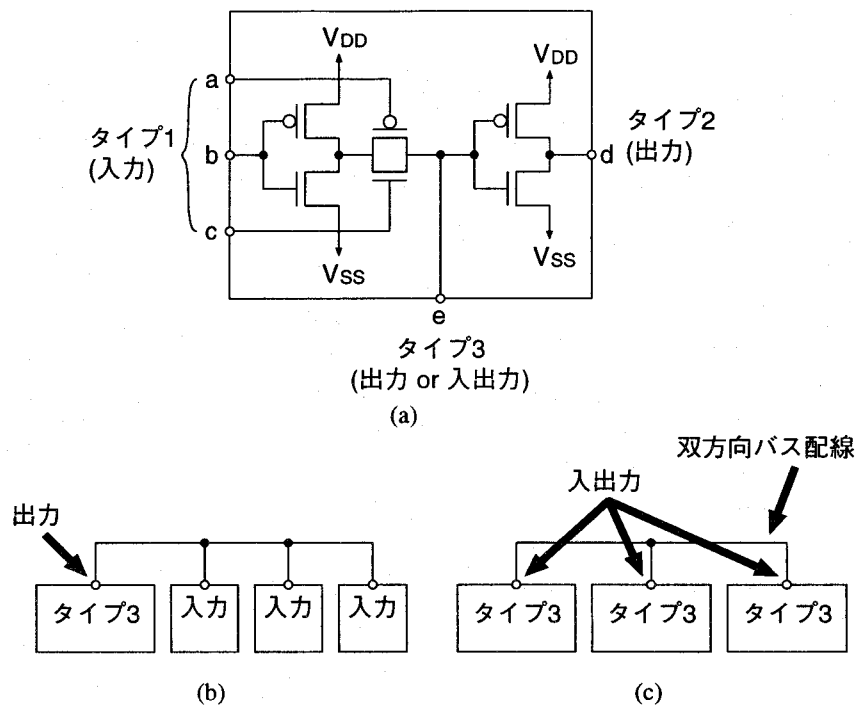


図 4.3 (a) 第一段階での端子の分類、(b) 第二段階で出力端子と判定される例、(c) 第二段階で入出力端子と判定される例

第二段階では、タイプ 3 の端子を出力端子と入出力端子に分類する。現在注目している端子を含めて、複数のタイプ 3 の端子が配線に接続していれば、それら全ては入出力端子である。配線に接続しているタイプ 3 の端子が現在注目している端子のみで、残りの端子が全て入力の場合、このタイプ 3 の端子は出力である。第二段階の分類の例を図 4.3 (b)、(c) に示す。図 4.3 (b) は出力と判定される例、図 4.3 (c) は入出力と判定される例である。

4.3.3 DC パスラベルによる上流追跡

MOS トランジスタが ON 状態の時、ソース端子とドレイン端子の間には、双方向に信号が流れることができる。このため、トランジスタレベルの故障追跡において問題が生ずる。レイアウトから抽出されたトランジスタレベル回路データには、信号の流れの向きの情報は明示的には示されていない。このことが、故障信号を上流

へ追跡することを困難にする。

例として、図 4.4 に示す CMOS 回路を考える。0, 1, ..., 10 は配線番号、a, b, ..., h は MOS トランジスタ名である。ここで、故障信号が配線 2 の上流から、配線 3, 9 へと伝搬している場合を考える。この故障信号を配線 9 から上流へ故障追跡する。追跡を進め、配線 3 まで来た時、2 つの追跡方向が考えられる。配線 2 の方向と配線 4, 5, 8 の方向である。後者の方向に追跡を進めた場合、誤った故障箇所を出力してしまう。

図 4.4 の回路を回路接続グラフに変換したものを図 4.5 に示す。ここで、配線は○、MOS トランジスタは□、ソース/ドレイン端子は実線の枝、ゲート端子は破線の枝で示されている。配線 3 まで追跡が進んだ時、配線 4, 5, 8 は追跡対象から除外すべきである。追跡すべき方向は、MOS トランジスタ a, b のゲート配線である配線 2 である。これらの MOS トランジスタは図 4.5 で太線で示された経路上にある。この太線は、追跡開始配線 9 と電源配線 V_{DD} , V_{SS} を結ぶ DC パスを表している。

従って、トランジスタレベルの故障追跡を開始する時、また MOS トランジスタ

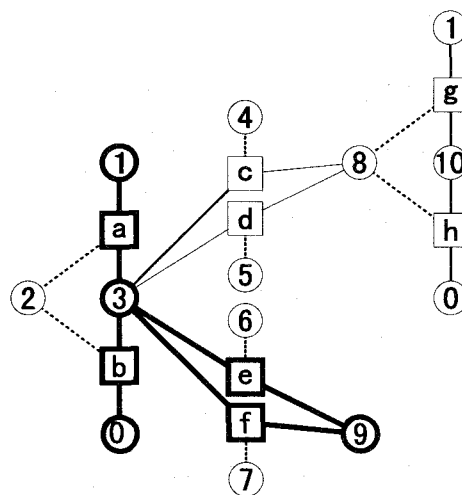
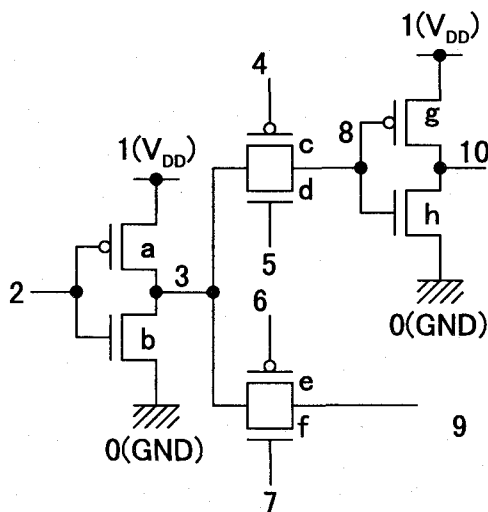


図 4.4 トランジスタレベル回路データ

図 4.5 図 4.4 の回路の回路接続グラフへの変換

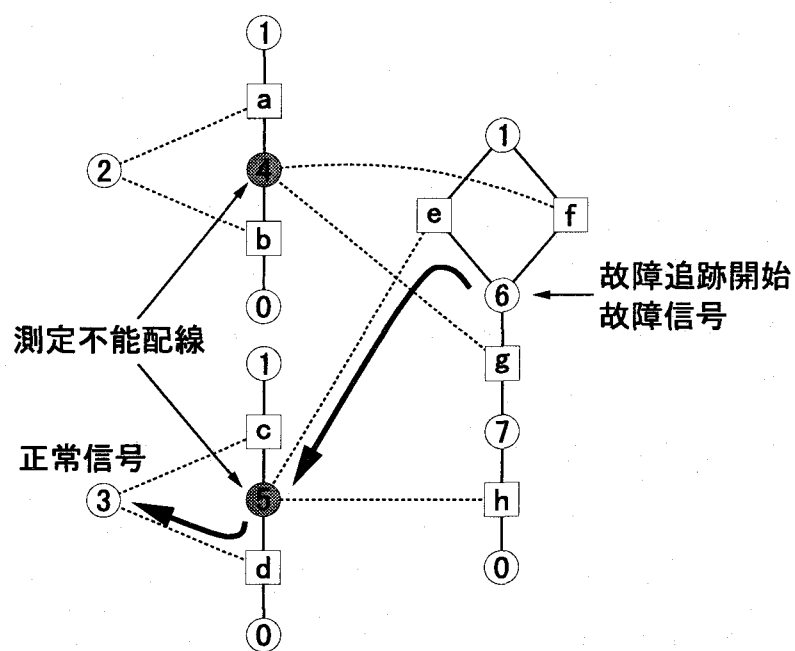
のゲート端子を遡る時に、注目配線から DC パスの探索を行い、DC パス上の配線および MOS トランジスタに DC パスであることを示すラベル (以後、DC パスラベルと呼ぶ) を付け、この DC パスラベルの付けられた経路に沿って上流追跡を行う。

階層構造レイアウトデータにおいて、セルとトランジスタが混在して記述されている階層レベルが存在する場合があるので、最下層のトランジスタレベルだけでなく、上位の階層においても DC パスラベルのラベル付けを行った上で、ラベルの付けられた経路に沿って追跡を行う。例えば、図 4.1、図 4.2 の回路において、配線 11 から追跡を行う場合、図 4.2 に太線で示した経路上の配線、MOS トランジスタに DC パスラベルが付けられる。

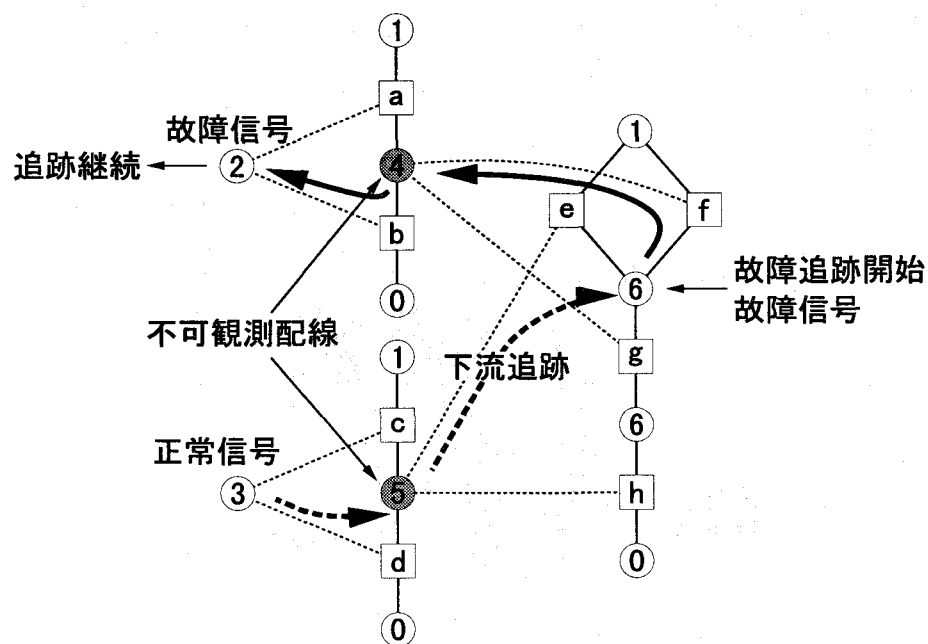
4.4 測定不能配線への対処

4.4.1 測定不能配線での下流追跡

EB テスタでは、第 2.4.1 節で述べたように、絶縁保護膜に覆われた下層配線上の電圧を測定する場合、電圧測定精度が大幅に低下する。このため、故障追跡中にこのような測定不能配線があった場合は、その配線は測定せずに、その上流の測定可能な配線を測定する。これが正常信号であった場合は、最後に故障信号が測定された下流配線に戻って、別の未測定の上流配線を測定する。この様子を図 4.6 に示す。ここで、故障は配線 2 の上流にあり、故障信号が $2 \rightarrow 4 \rightarrow 6 \rightarrow \dots$ と伝搬している場合を考える。この故障信号を配線 6 から追跡する時、追跡方向としては、配線 4 の方向と配線 5 の方向の二つの方向がある。まず、配線 5 の方向が選択された場合、図 4.6(a) に示されているように、測定不能配線 5 は飛ばして、その上流の測定可能な配線である配線 3 を測定する。配線 3 では正常信号が測定される。正常な信号が測定された場合、図 4.6(b) に点線の矢印で示されているように、最後に故障信号が測定された配線 6 まで下流追跡し、その後実線の矢印で示されているように、配線 4, 2 の方向に追跡を進める。



(a)



(b)

図 4.6 測定不能配線での下流追跡

4.4.2 下流部分および測定不能配線に存在する故障の扱い

本アルゴリズムにより測定対象から除外される下流部分に故障が存在する可能性も考えられる。例えば、図 4.4 の回路において配線 8 が V_{DD} とショートしており、MOS トランジスタ c, d が ON 状態の場合には、故障信号が配線 3 へ逆流し、配線 9 へと伝搬する。このような故障を配線 9 から故障追跡すると、配線 3 まで追跡され、配線 2 では正常信号が測定されることから、配線 3 または MOS トランジスタ a, b に故障があると結論づけられる。

また、測定不能配線上に故障が存在する場合も考えられる。

以上のようなことから、故障追跡が終了し、最後に故障信号が測定された配線に測定不能配線や DC パスラベルが付けられていない配線（即ち、測定されない下流配線）が隣接している場合は、それらも故障の可能性のある配線として警告を表示する。この例を図 4.7 に示す。配線 10 から故障追跡を進め、配線 4 では故障信号、配線 2 では正常信号が測定された場合を考える。配線 3 は下層配線のため測定不能である。このとき、図 4.7 で網かけされている配線 3, 4, 9、MOS トランジスタ a, b, c, d, e, f が故障の可能性のある箇所として指摘される。

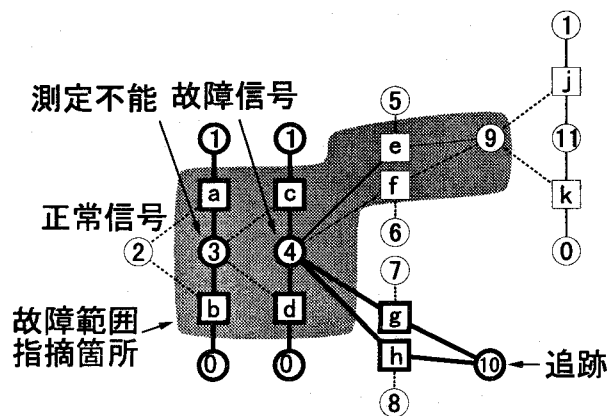


図 4.7 測定不能配線下に故障がある場合の故障箇所指摘範囲

4.4.3 FIB・EB 統合化システムによる、測定不能配線の故障箇所絞り込み

測定不能配線を測定する手法として、FIB (Focused Ion Beam, 集束イオンビーム) 加工装置を用いて絶縁膜に穴を空け、タングステン等の金属を堆積させて EB 測定用パッドを作成する方法が報告されている^{[50],[51]}。FIB 加工装置と EB テストシステムが別々の装置である場合、加工・測定の度に試料装着・真空引きを行う必要があり、非常に時間がかかる。このため、図 4.8 に示されているように、一つの試料室に FIB 鏡筒と EB 鏡筒を統合した FIB・EB 統合化装置が報告されている^{[52]~[54]}。このような装置を用いれば、絶縁膜下の配線も正確に測定することができ、より正確に故障箇所を絞り込むことが可能である。しかしながら、FIB での加工は時間を要するので、加工箇所はなるべく少ないことが望まれる。

以上のようなことから、まず第 4.4.2 節で示したように EB テストシステムのみで可能な範囲まで絞りこみ、FIB が使用可能ならば、続いて FIB によるパッド作成と EB テストシステムによる測定を行って、より故障箇所を絞り込む。

FIB・EB 統合化システムが利用可能な場合には、上述したように二段階の故障追跡を行なうことにより、故障箇所をより正確に、かつ効率的に絞り込むことができる。

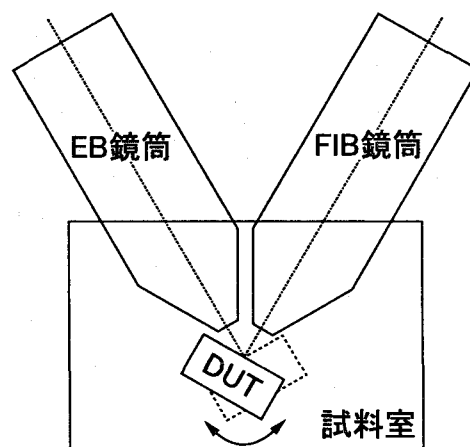


図 4.8 FIB・EB 統合化装置の概略

4.5 順序回路の故障追跡

4.5.1 測定位相の制御

順序回路において性能故障の追跡を行う場合、第2.4.2節で述べたように、EBテストシステムでの波形測定時間が必然的に長くなる。測定時間を短くするためには、故障信号の測定される位相範囲についてのみ波形の測定を行う必要がある。特に、故障信号が生じ始める位相が重要である。故障信号が生じ始める位相は、故障信号の伝搬とともに遅れていく。組合せ回路ブロックでの信号の遅延はシステムクロック周期に比べて小さいが、メモリ要素での信号の遅延はクロック周期と同程度か、それよりも大きい。このことから、次の条件を満たすように、測定位相の制御を行うことが重要である。

- (1) 測定位相範囲はなるべく短いことが望まれる。
- (2) 故障信号が生じ始める位相が測定位相範囲内に入っていないといけない。

測定開始位相の制御は、次のように行う。最後に行った測定での故障信号の開始位相を t_0 とする時、今回の測定の開始位相を $t_0 - T_A$ とする。 T_A は定数で、上述の条件を満たすように、クロック周期の数倍とする。

測定終了位相は、故障信号の種類に依存して変化する。我々の手法は縮退故障だけでなく、性能故障を取り扱う。性能故障では、伝搬遅延、立ち上がり・立ち下がり時間などの波形の時間的パラメータが問題となる。そこで、故障信号の種類を次の3つに分類する。

- (1) DUT と良品 VLSI の信号遅延の差が、あらかじめ定めた量を上回っている。
- (2) DUT と良品 VLSI の信号の立ち上り（立ち下がり）時間の差が、あらかじめ定めた量を上回っている。
- (3) 論理レベルが異なる。

(1), (2) の場合には、立ち上がり（立ち下がり）エッジが測定位相範囲内に入っていないといけない。そこで、図4.9に示すように、測定の終了位相を $t_1 + T_B$ とす

る。ここで、 t_1 はDUTの波形が振幅の90%（10%）を横切る位相である。定数 T_B は、クロック周期の1/4程度にする。(3)の場合には、図4.10(a)に示すように、測定を終了位相を $t_1 + T_B$ とする。ここで、 t_1 は、DUTの波形と基準波形の差異が無くなる位相である。但し、 $t_1 + T_B - t_0 > T_C$ となる場合は、図4.10(b)に示すように、測定を終了位相は $t_0 + T_C$ とする。ここで、 T_C は、クロック周期と同程度の定数である。

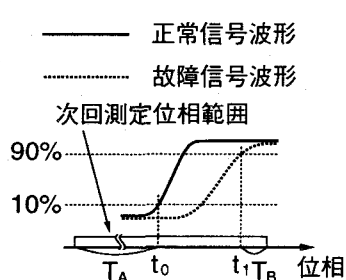


図 4.9 性能故障の場合
の測定位相の決定

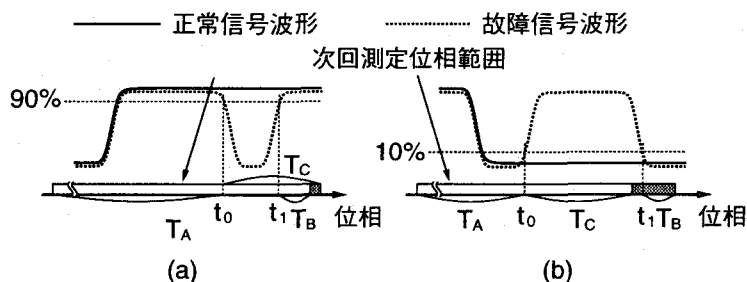


図 4.10 論理故障の場合の測定位相の決定

4.5.2 異なる位相における同一配線のテスト

順序回路中には様々なフィードバックループが存在する。これらのループは2種類に分類することができる。

- (1) 信号がループを循環するのにかかる時間がシステムクロック周期以上のもの。
フリップフロップ等のメモリ要素と組合せ回路ブロックの組合せからなる、図4.11に示すようなループ。ここで、図中のFFはフリップフロップを表している。
- (2) 信号がループを循環するのにかかる時間がシステムクロック周期未満のもの。
例えば、メモリ要素内部等に見られる小さいループ。

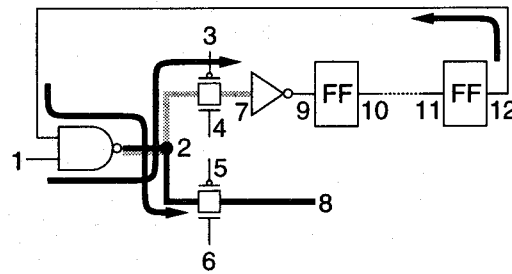


図 4.11 順序回路中でのフィードバックループの例

上述 (1) のループでは、一度ある配線を伝搬した後、故障信号もクロック周期以上の時間遅延して、再度同じ配線を伝搬する可能性がある。例えば、図 4.11 で太線の矢印で示したように故障信号が伝搬する。第 4.5.1 節で述べたように、本手法では故障信号が測定される数クロック周期の間しか波形測定を行わないので、この様に故障信号が複数回同じ配線を伝搬している場合、複数回同じ配線を通して追跡しなければならない。

上述 (2) のループを故障信号が伝搬している場合、(1) の場合とは異なり、同じ配線を複数回追跡することを許すと、追跡が無限にループを回ってしまう可能性がある。

上述の問題を避ける為、EB テストシステムで測定を行った配線には、目印としてラベルを付け、付加情報として測定を行った位相範囲を記録する。このラベルを測定済みラベルと呼ぶ。測定済みラベルに記録された測定位相範囲と、これから行おうとする測定の位相範囲に重複がなければ、新たに測定を行う。重複がある場合は、再測定は行わず、他の未測定の上流配線を測定する。

この他にも、DC パスラベルに関する問題がある。図 4.11 の回路で故障追跡が配線 7 まで進んだ時に行なわれる DC パスラベル付けでは、灰色の太線で示されている配線およびこの配線上の MOS トランジスタに DC パスラベルが付けられる。しかし、黒い太線で示されている配線 2, 8 およびこの配線上の MOS トランジスタには、既に 1 回目の追跡で DC パスラベルが付けられている。これを無視して新たに DC パスラベルを付けた場合、故障追跡が誤った方向、即ち、配線 5, 6, 8 の方向に進む可能性がある。

この問題を解決する為、DC パスラベルに付加情報として探索深さを記録する。探索深さは、故障追跡開始時は 1 で、MOS トランジスタのゲート端子やセルを遡って追跡を進める時には+1 される。既に DC パスラベルが付けられている配線でも、探索深さが異なれば新たに DC パスラベルを付けることができる。

4.6 双方向バス配線での故障追跡

双方向バス配線の基本的な構造を図 4.12 に示す。双方向バスに接続されている各セルは、入力バッファと 3 状態出力バッファ (以降では、トライステート出力バッファと呼ぶ) を有している。トライステート出力バッファにはデータ入力と制御入力がある。制御入力は、バッファの出力をハイインピーダンス状態にするか、データ入力を出力へ伝えるかを決定する。図 4.12 のトライステート出力バッファの場合、制御入力が接地電位 (“L”) の時ハイインピーダンスになり、制御入力が電源電圧 (“H”) の時データ入力が出力へ伝えられる。ある瞬間に双方向バスに対してデータを出力することができるセルは一つだけである。この様に、入力、出力の両方の機能を果たす端子を入出力端子と呼ぶ。

故障追跡時には、信号の伝搬方向を知る必要がある。しかしながら、双方向バスにおいては、信号の流れの向きが時々刻々変化する。従って、注目する測定位相範

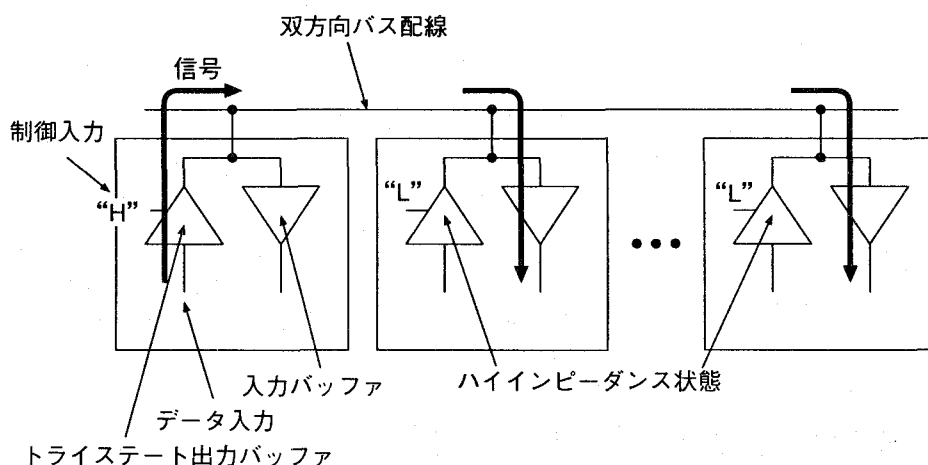


図 4.12 双方向バス配線の基本構造

囲における信号の流れの向きを把握する必要がある。

4.6.1 入出力端子における信号の流れの向き

故障追跡が双方向バス配線に到達した時、信号の流れの向きを制御しているトライステート出力バッファの回路を認識し、その制御入力を測定することで、その時点における信号の流れの向きを知ることができる。しかしながら、このような手法では、回路認識を行うのに時間がかかる。また、カスタム設計の LSI で、特殊な回路構成のトライステート出力バッファを用いていた場合、認識に失敗してしまう。

そこで、回路機能の認識を行わずに故障追跡を行う手法を提案する。トライステート出力ドライバーが故障信号を出力している時 (出力ドライバーの内部に故障がある場合を除けば)、データ入力か制御入力のいずれかで故障信号が測定されるはずである。従って、出力ドライバーの入力配線 (制御入力、データ入力ともに) を測定し、故障信号が検出されたドライバーが信号を出力していると考えればよい。

4.6.2 双方向バス配線での測定位相

双方向バス回路では、しばしば、故障信号が双方向バス配線を伝搬し、あるセル内に入った後に遅延して、再び双方向バス配線を伝搬することがある。このため、同一の双方向バス配線上の異なる位相において、何度も同じ故障信号が測定される。このような故障信号の追跡を、他の回路部分と同じように第 4.5.1 節で示した測定位相の制御手法で追跡したのでは、追跡効率が低下してしまう。

このようなことから、双方向バス配線においては、十分に長い位相範囲の波形測定を行った方が効率的と考えられる。双方向バス配線を測定する場合、本手法では、テストシーケンスの先頭から、現在注目している位相までの波形測定を行う。

4.6.3 双方向バス配線での故障追跡法

双方向バス回路での故障追跡法を、図 4.13 の回路を例に説明する。

図において 1, 2, ..., 18 は配線番号。A, B, C はセル名である。かぎ括弧 [] で囲ま

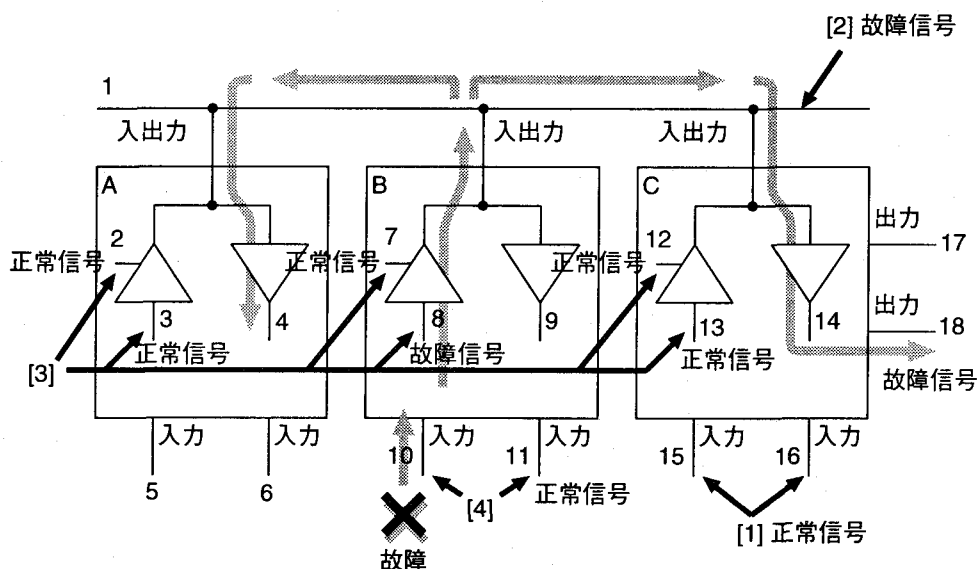


図 4.13 双方向バス回路での故障追跡法

れた数字は、配線の測定順序をあらわしている。×印の位置に故障があり、故障信号は灰色の矢印で示したように伝搬している。

この故障信号を配線 18 から追跡した場合、まず単方向の入力配線である配線 15, 16 を測定し (測定 [1])、正常な信号が測定される。次に、双方向バス配線 1 を測定する (測定 [2])。ここで、測定位相は、入力テストシーケンスの先頭から、配線 18 において故障信号が測定された位相までとなる。ここでは故障信号が測定される。次にセル A, B, C の出力ドライバーの入力配線である配線 2, 3, 7, 8, 12, 13 が測定候補となり、これらが順次測定される (測定 [3])。このうちで、配線 8 において故障信号が測定される。これにより、セル B が故障信号を出力していることが分かる。続いて、セル B の入力配線である配線 10, 11 が測定候補となり、順次測定される (測定 [4])。配線 10 で故障信号が測定され、配線 10 から通常の故障追跡が行われる。

4.7 アルゴリズムの詳細

階層的故障追跡アルゴリズムの詳細を以下に示す。なお、以下の説明で、探索深さ i の DC パスラベルを $\lambda_D(i)$ 、位相範囲 $[t_1, t_2]$ の測定済みラベルを $\lambda_m(t_1, t_2)$ と表

す。また、探索深さや位相範囲を特定せず、MOS トランジスタおよび配線に付けられた全ての DC パスラベル、測定済みラベルを指す場合は、それぞれ、単に λ_D 、 λ_m と表す。

故障追跡手続き

ステップ 1: 全ての配線および MOS トランジスタに付けられた λ_D 、 λ_m を消去する。

ステップ 2: 故障追跡を開始する配線 s から DC パスの探索を行い、DC パス上の配線および MOS トランジスタに $\lambda_D(1)$ を付ける。

ステップ 3: 故障追跡開始配線に $\lambda_m(0, t_e)$ (t_e はテストシーケンスの終了位相) を付ける。

ステップ 4: LSI テスタで測定された故障信号の位相範囲と第 4.5.1 節の測定位相制御法より次の測定の測定位相を決定し、これを $[t_1, t_2]$ とする。

ステップ 5: 変数 m を “without-FIB mode” とする。

変数 m は、FIB を使用していないことを意味する “without-FIB mode” と FIB の使用を意味する “with-FIB mode” のいずれかの値を持つ。故障追跡の初めは “without-FIB mode” であり、EB テスタのみでの故障追跡が終了して FIB 加工装置利用する場面が来ると、“with-FIB mode” になる。

ステップ 6: 関数 *searchFault* を呼び出す: *searchFault*($s, [t_1, t_2], 1$)

関数 *searchFault*($v, [t_1, t_2], i$) は、パラメータとして注目配線 v 、測定位相 $[t_1, t_2]$ 、探索深さ i を与えることにより、配線 v より故障追跡を行う。配線 v およびその上流配線で故障箇所が見つかった場合は、返却値として偽を返し、故障箇所が見つからなかった場合は真を返す。また、関数 *searchFault* は、測定不能配線での下流追跡を行う為、再帰呼び出しを用いた深さ優先探索を行う。注目配線が一段上流へ移る度に、新たに関数 *searchFault* が呼び出される。逆に測定不能配線のために下流追跡を行う場合は、注目配線が一段下流へ戻る度に、呼び出された時とは逆の順番で関

数 *searchFault* が終了する。故障箇所が同定されると、全ての関数 *searchFault* が終了し、処理が終わる。

関数 *searchFault* の詳細を以下に示す。また、処理のフローチャートを図 4.14 に示す。

関数 *searchFault*($v, [t_1, t_2], i$)

ステップ F1: 波形測定、比較

このステップでは、配線 v の波形測定等が行われる。まず、配線 v が測定済みであるか否かをチェックする。 v に λ_m が付けられていないか、付けられている場合でもその位相と $[t_1, t_2]$ とに重なりがなければ、未測定である。

v が測定済みの場合はステップ F2 へ移る。この処理は、故障追跡開始時に追跡開始配線の測定を省略したり、故障箇所を含むセルが同定されてセル内部の追跡に移る時に、セルの出力端子に接続されたセル内部の配線の測定を省略する為に行う。故障追跡開始する配線は、LSI テスタにより既に測定が行われているので、再測定を行う必要はない。また、故障箇所を含むセルが同定されてセル内部の追跡に移る時も、セルの出力端子に接続されているセル内部の配線は測定する必要はない。なぜなら、この時、セルの出力端子に接続されているセル外の配線は既に測定済みであり、この端子に接続されているセル内の配線は、電気的にはセル外の配線と同一だからである。故障追跡開始時には、故障追跡手続きのステップ 3 で故障追跡開始配線に測定済みラベルが付けられる。また、故障箇所を含むセルが同定されてセル内部の追跡に移る時には、後述するステップ F7 でセルの出力端子に接続されたセル内部の配線に測定済みラベルが付けられる。これらの測定済みラベルによって、これらの配線の測定が省略される。

v が未測定の場合は v が測定可能であるか否かをチェックする。この判定は、変数 m の値に依存する。 $m = \text{without-FIB mode}$ の場合は、 v の構成ポリゴンに最上層金属配線のポリゴンが含まれていれば測定可能と判定する。 $m = \text{with-FIB mode}$ の場合は、配線構成ポリゴンに金属層のポリゴンが含まれ、か

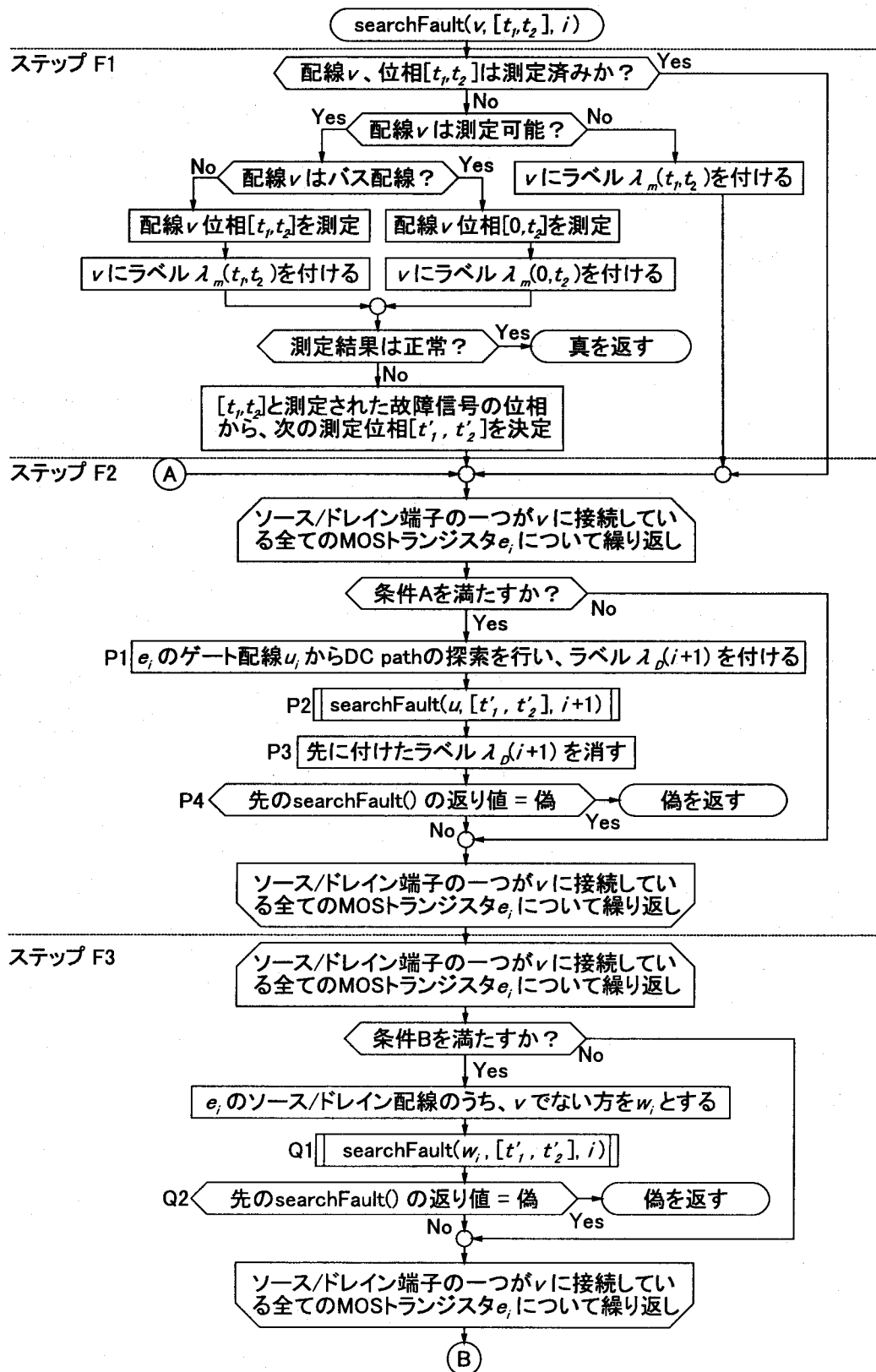
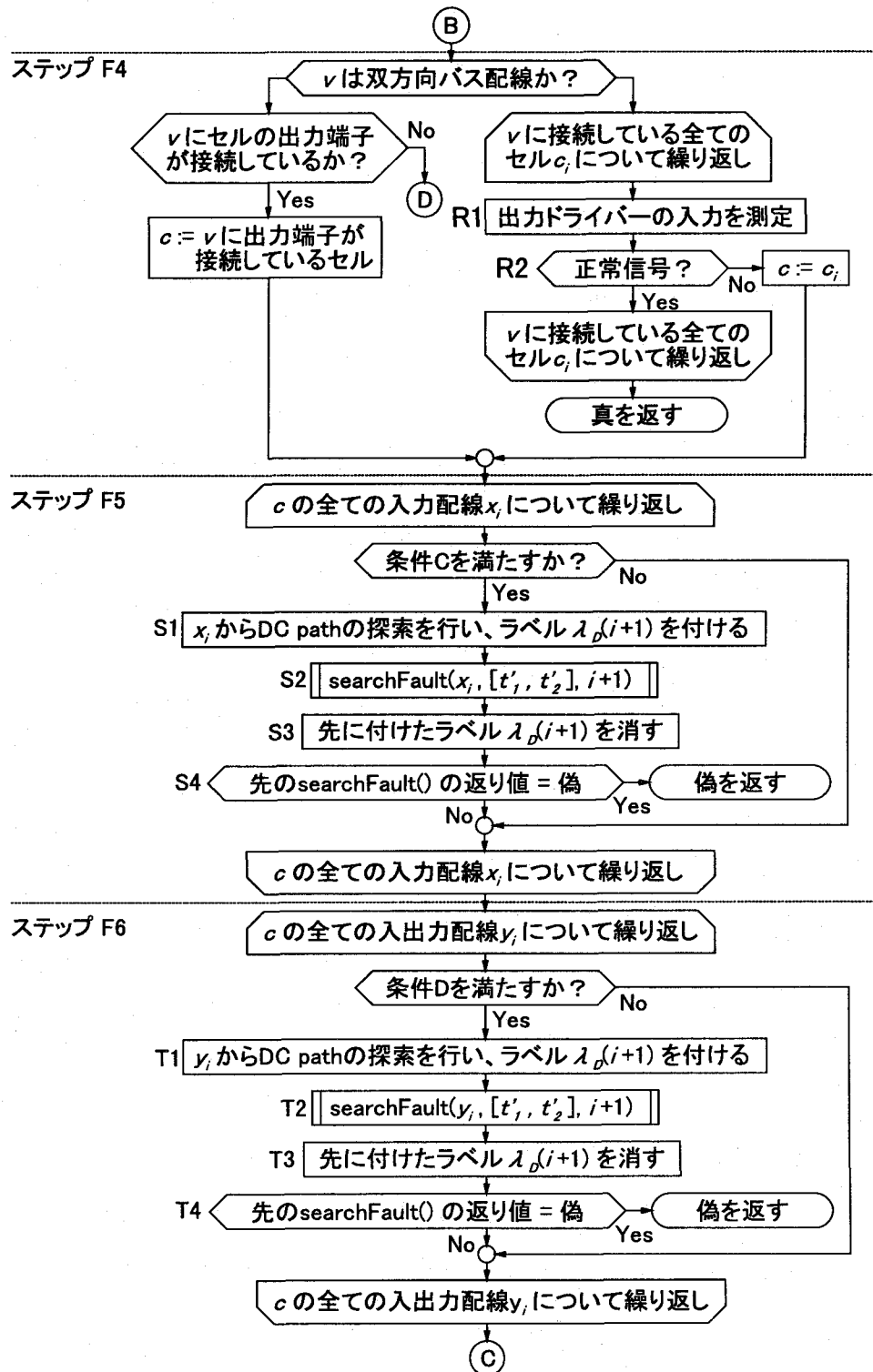
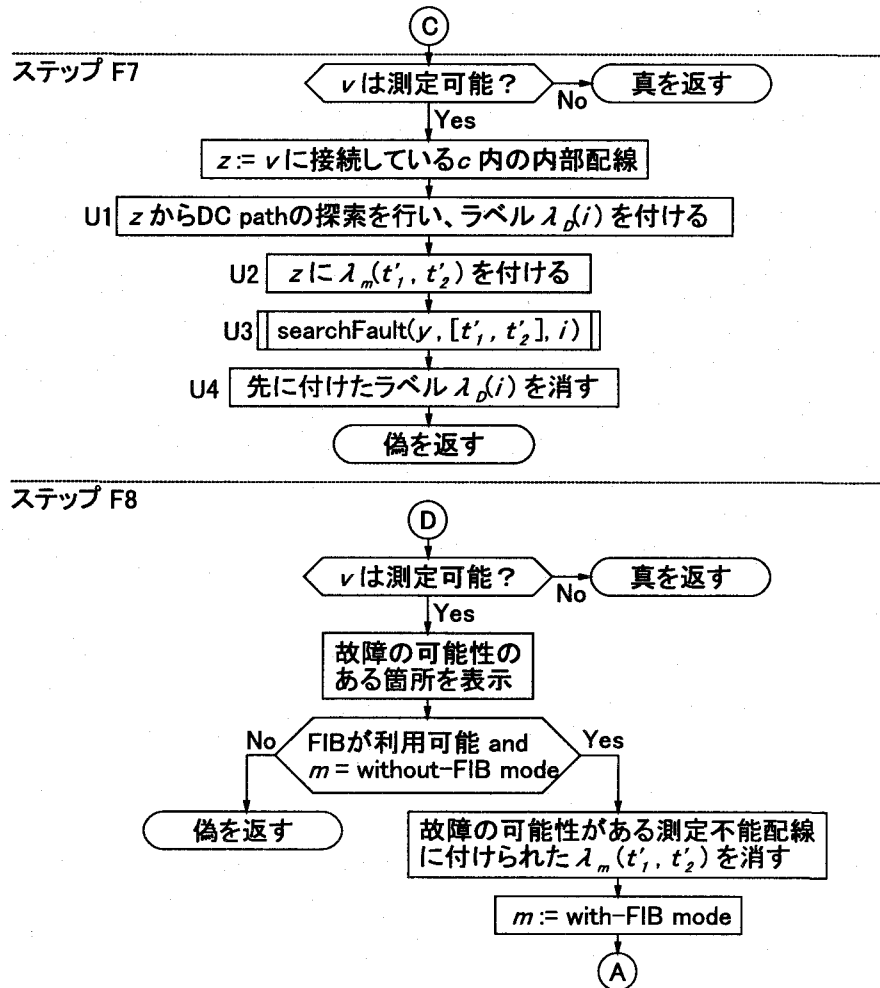


図 4.14 関数 searchFault の詳細 (1/3)

図 4.14 関数 *searchFault* の詳細 (2/3)

図 4.14 関数 `searchFault` の詳細 (3/3)

つそれらに上層の金属配線で覆われていない部分が存在すれば、測定可能と判定する。

v が測定不能の場合には、 v に $\lambda_m(t_1, t_2)$ を付け、ステップ F2 へ移る。測定を行っていないにも関わらず λ_m を付加するのは、回路中にフィードバックループが存在し、そのループ上の全ての配線が測定不能であった場合に、ループを無限に回ってしまうことを防止する為である。

v が測定可能な場合は、 v が双方向バス配線であるか否かを判定する。 v にセルの入出力端子が接続されていれば、 v は双方向バス配線である。 v が双方向バス配線の場合は、 v の位相範囲 $[0, t_2]$ を測定し、 v に $\lambda_m(0, t_2)$ を付ける。 v

が通常配線の場合は、 v の位相範囲 $[t_1, t_2]$ を測定し、 v に $\lambda_m(t_1, t_2)$ を付ける。

この測定結果が正常信号であれば、関数 *searchFault* を終了し、真を返す。
故障信号の場合は、第 4.5.1 節に示した方法で、 $[t_1, t_2]$ から次の測定位相 $[t'_1, t'_2]$ を決定し、ステップ F2 へ移る。

ステップ F2: MOS トランジスタのゲート配線のテスト

このステップでは、 v にソース/ドレイン端子が接続されている MOS トランジスタのゲート配線をテストする。配線 v にソース/ドレイン端子が接続されている MOS トランジスタを e_i ($i = 1, 2, \dots, 3$) とし、各 e_i に対し、次の条件 A を満たせば、後述する手続き P を実行する。

条件 A

- (A1) e_i に $\lambda_D(i)$ が付けられている。
- (A2) e_i に λ_m が付けられていないか、または λ_m が付けられており、その位相範囲と $[t'_1, t'_2]$ とに重なりがない。
- (A3) e_i のゲート配線を u_i とし、 u_i が電源配線ではない。
- (A4) e_i のゲート配線 u_i から DC パスが存在し、その DC パス上の配線および MOS トランジスタに $\lambda_D(i)$ が付けられていない。

手続き P

- (P1) u_i からの DC パス上の配線および MOS トランジスタに、 $\lambda_D(i+1)$ を付ける。
- (P2) 関数 *searchFault* を呼び出す: *searchFault*($u_i, [t'_1, t'_2], i+1$)
- (P3) (P1) で付けられた $\lambda_D(i+1)$ を消去。
- (P4) (P2) で呼び出した関数 *searchFault* の返却値が偽なら、現在実行中の関数 *searchFault* を終了し、偽を返す。

ステップ F2 の例を図 4.15 に示す。配線 s から故障追跡を開始し、太い矢印で示すように追跡が進められ、配線 6 まで追跡が到達した場合を考える。ここで、配線 8, 9, 10, 11 は故障追跡とは関係のない信号の下流部分である。配線 6 から DC パスラベルを付ける操作が行なわれ、太線で示されている枝上の MOS トランジスタ e, f, g, h および配線 6, 7 に $\lambda_D(i)$ が付けられた後、配線 6 をパラメータとして、関数 *searchFault* が呼び出される。配線 6 にソースドレイン端子が接続されている MOS トランジスタは e, f, g, j, k であり、このうち、条件 A を満たすのは e, f, g のみである。従って、これらのゲート配線 4, 5 をパラメータとして関数 *searchFault* が呼び出される。配線 4, 5 のいずれかで故障信号が見つければ、その方向に追跡が進められた後偽が返されるので、現在呼び出されている関数 *searchFault* を終了し、偽を返す。配線 4, 5 の両方で正常信号が測定された場合は、真が返されるので、ステップ F3 へ移る。

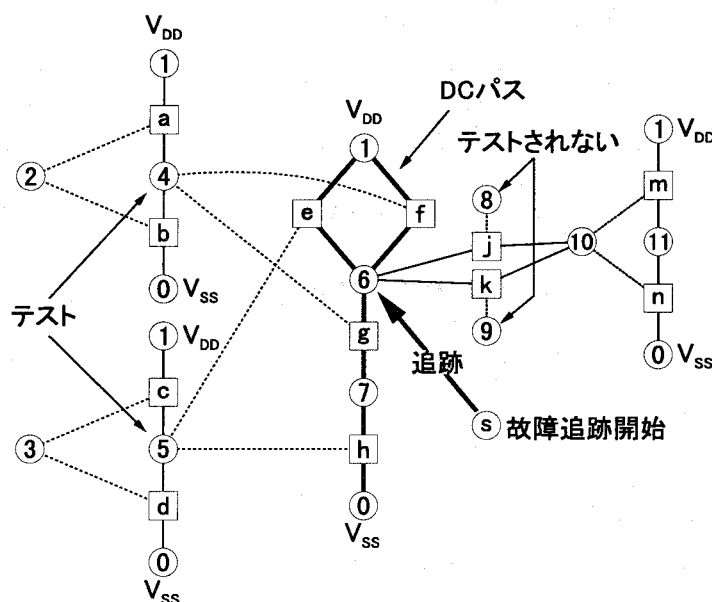


図 4.15 関数 *searchFault* のステップ F2 の例

ステップ F3: MOS トランジスタのソース/ドレイン配線のテスト

このステップでは、MOS トランジスタのチャンネルを介して v と接続されている配線をテストする。配線 v にソース/ドレイン端子が接続されている MOS

トランジスタを e_i とし、各 e_i に対し、次の条件 B を満たせば、後述する手続き Q を実行する。

条件 B

- (B1) e_i の 2 つのソース/ドレイン配線のうち v でないものを w_i とし、 w_i に DC パスラベル $\lambda_D(i)$ が付けられてる。
- (B2) w_i は電源配線ではない。
- (B3) w_i に λ_m が付けられていないか、または λ_m が付けられており、その位相範囲と $[t'_1, t'_2]$ とに重なりがない。

手続き Q

- (Q1) 関数 *searchFault* を呼び出す: *searchFault*($w_i, [t'_1, t'_2], i$)
- (Q2) (Q1) で呼び出した関数 *searchFault* の返却値が偽であれば、現在呼び出されている関数 *searchFault* を終了し、偽を返す。

ステップ F3 の例を、図 4.16 に示す。図 4.15 と同じく、配線 s から故障追跡を開始し、太い矢印で示すように追跡が進められ、配線 6 まで追跡が到達した場合を考える。 $\lambda_D(i)$ は、図 4.15 に太線で示されている枝上の MOS トランジスタ e, f, g, h および配線 6, 7 に付けられている。配線 6 にソース/ドレイン端子が接続されている MOS トランジスタは e, f, g, j, k であり、このうち、条件 B を満たすのは g のみである。従って、配線 7 のみがテストされる。

全ての出力ドライバの入力配線で正常信号が測定された場合は、関数 *search-Fault* を終了し、真を返す。

ステップ F5: セルの入力配線のテスト

このステップでは、故障信号を出力しているセル c の入力配線をテストする。 c の全ての入力配線を x_i とし、各 x_i に対して、条件 C を満たせば、後述する手続き S を実行する。

条件 C

- (C1) x_i に λ_m が付けられていないか、または λ_m が付けられており、その位相範囲と $[t'_1, t'_2]$ とに重なりがない。
- (C2) x_i からの DC パスが存在し、その DC パス上の配線および MOS トランジスタに $\lambda_D(i+1)$ が付けられていない。

手続き S

- (S1) x_i からの DC パス上の配線および MOS トランジスタに、DC パスラベル $\lambda_D(i+1)$ を付ける。
- (S2) 関数 *searchFault* を呼び出す: *searchFault*($x_i, [t'_1, t'_2], i+1$)
- (S3) (S1) で付けられた $\lambda_D(i+1)$ を消去。
- (S4) (S2) で呼び出した関数 *searchFault* の返却値が偽なら、現在実行中の関数 *searchFault* を終了し、偽を返す。

ステップ F6: セルの入出力配線のテスト

このステップでは、故障信号を出力しているセル c の入出力配線をテストする。 c の全ての入出力配線を y_i とし、各 y_i に対して、条件 D を満たせば、後述する手続き T を実行する。終了後、ステップ F7 へ移る。

条件 D

- (D1) y_i に λ_m が付けられていないか、または λ_m が付けられており、その位相範囲と $[t'_1, t'_2]$ とに重なりがない。
- (D2) y_i からの DC パスが存在し、その DC パス上の配線および MOS トランジスタに $\lambda_D(i+1)$ が付けられていない。

手続き T

- (T1) y_i からの DC パス上の配線および MOS トランジスタに、 $\lambda_D(i+1)$ を付ける。
- (T2) 関数 *searchFault* を呼び出す: *searchFault*($y_i, [t'_1, t'_2], i+1$)
- (T3) (S1) で付けられた $\lambda_D(i+1)$ を消去。
- (T4) (S2) で呼び出した関数 *searchFault* の返却値が偽なら、現在実行中の関数 *searchFault* を終了し、偽を返す。

ステップ F7: セル内部の故障追跡

このステップでは、故障箇所を含むセルの内部の追跡を行う。まず v が測定可能配線であるか、否かをチェックする。測定不能配線の場合は、 v の測定を飛ばして、その上流配線を測定し、それら全てにおいて正常信号が測定された場合である。この場合は、関数 *searchFault* を終了し、真を返す。測定可能配線の場合は、 v に接続している c 内の配線を z とし、以下の手続き U を実行する。

手続き U

- (U1) z からの DC パス上の配線および MOS トランジスタに、 $\lambda_D(i)$ を付ける。
- (U2) z に $\lambda_m(t'_1, t'_2)$ を付ける。
- (U3) 関数 *searchFault* を呼び出す: *searchFault*($z, [t'_1, t'_2], i$)

(U4) (U1) で付けられた $\lambda_D(i)$ を消去。

(U5) 関数 *searchFault* を終了し、偽を返す。

ステップ F8: 故障箇所を表示

このステップでは、故障箇所の表示等を行う。まず、 v が測定可能であるか、否かをチェックする。測定不能の場合は、関数 *searchFault* を終了し、真を返す。 v が測定可能な場合は、 v かその周辺に故障があるので、第 4.4.2 節で示した範囲に存在する配線および MOS トランジスタを故障の可能性のある箇所として表示する。次に、 $m = \text{without-FIB mode}$ であり、かつ FIB が利用可能な場合は、故障の可能性のある測定不能配線に付けられた $\lambda_m(t'_1, t'_2)$ を消去し、 $m := \text{with-FIB mode}$ として、ステップ F2 へ移る。これにより、以後は下層金属層のみからなる配線も測定可能配線として扱われ、FIB により絶縁膜への穴開け加工、パッド作成を行った上で、波形測定が行われる。FIB が利用不可能な場合、あるいは既に $m = \text{with-FIB mode}$ の場合は、故障追跡が終了したので、関数 *searchFault* を終了し偽を返す。

4.8 評価

故障追跡アルゴリズムの有効性を示すため、本アルゴリズムおよび第 3 章で示した CAD レイアウトからの逐次回路抽出法を UNIX ワークステーション上で C 言語を用いて実装し、商用 VLSI の回路データおよびモデルレイアウトデータへ適用した。本節ではこれらの結果を示す。

4.8.1 トランジスタレベル故障追跡

DC パスを用いたトランジスタレベルの故障追跡アルゴリズムの有効性を示すため、トランジスタレベル故障追跡の部分のプログラムを商用 DRAM 評価用回路の部分回路 (CMOS、約 5k トランジスタ) に適用した。実際に LSI の故障追跡を行う場合には、DUT 波形と基準波形 (シミュレーション波形もしくは良品測定波形) と

を比較して、信号が正常であるか否かを判断するが、本プログラムでは、あらかじめ故障信号が伝搬している配線を設定してファイルに記録しておき、このファイル参照して故障追跡を行なう。本プログラムでは、トランジスタレベルの回路データとして、SPICE 形式のネットリストが利用可能である。

本プログラムは、SPICE 形式のネットリストを読み込んで回路データ構造をメモリー上に構成した後、故障信号の伝搬している配線を設定したファイルを読み込み、故障追跡を行う。本プログラムを図 4.17 に示す部分 (部分回路 1) について適用した。図中の数字は配線番号を表している。これをトランジスタレベルで表示したものが図 4.18 である。図中の数字は同じく配線番号を表しており、M に続く数字は MOS トランジスタの番号を表している。故障は配線 11 に設定されており、太線で示されている配線は、故障信号が伝搬した配線である。プログラムの出力結果を表 4.1 に示す。表中の “measure” で始まる各行は、プロービング点の番号、配線番号、測定

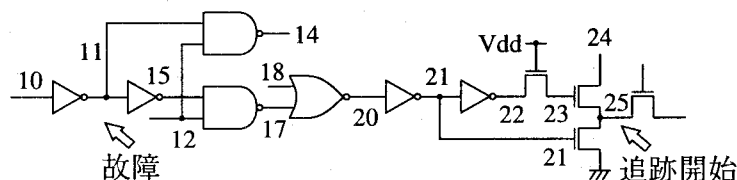


図 4.17 商用 DRAM の部分回路 1 のゲートレベル回路図

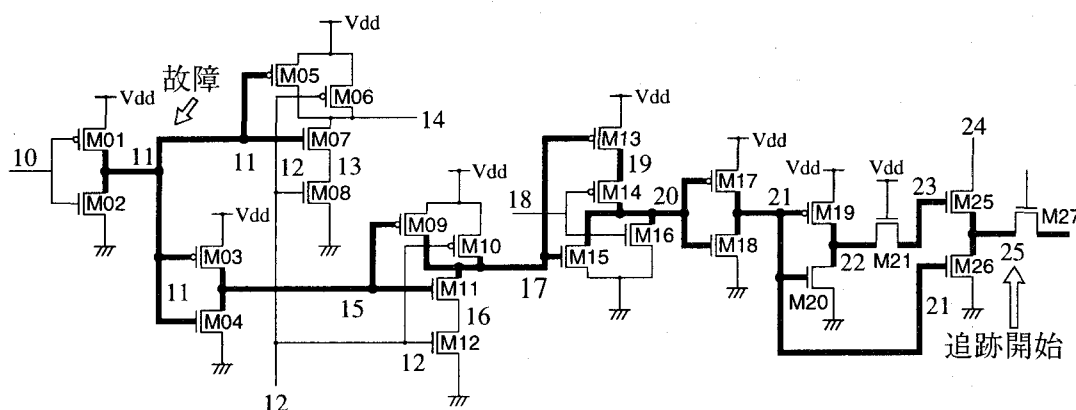


図 4.18 商用 DRAM の部分回路 1 のトランジスタレベル回路図

表 4.1 商用 DRAM の部分回路 1 に対するトランジスタレベル故障追跡の実行結果

1:	measure[1]:23, Faulty
2:	measure[2]:22, Faulty
3:	measure[3]:21, Faulty
4:	measure[4]:20, Faulty
5:	measure[5]:18, Good
6:	measure[6]:17, Faulty
7:	measure[7]:15, Faulty
8:	measure[8]:11, Faulty
9:	measure[9]:10, Good
10:	There may be a fault around the interconnection 11.

結果を示している。

まず、配線 25 に接続している MOS トランジスタ M25, M26 のゲート配線である 23, 21 が測定候補となる。このうち、どちらが初めに測定されるかは、回路データの記述の仕方やプログラムのコーディングに依存する。表 4.1 では先に配線 23 が測定され、故障信号が検出される。次に MOS トランジスタ M21 のゲート配線は電源なので測定されず、配線 22 が測定され、故障信号が検出される。同様にして、配線 21, 20 の測定が行われ、それぞれで故障信号が検出される。次の測定候補は、配線 20 に接続されている MOS トランジスタ M14, M15, M16 のゲート配線である配線 17, 18 である。表 4.1 では先に配線 18 が測定され正常信号が検出されたので、次に配線 17 が測定され、故障信号が検出される。さらに、配線 15, 11 が測定されて、それぞれで故障信号が検出される。その次の測定対象である配線 10 では正常信号が検出され、配線 11 か MOS トランジスタ M01, M02 に故障があることが判明する。このように、想定した故障が正しく特定された。

次に、本プログラムを同じ LSI の図 4.19 に示す部分 (部分回路 2) について適用した。図 4.20 はこれをトランジスタレベルで表示したものである。故障は配線 31 に設定されている。この結果を表 4.2 に示す。この場合も、想定した故障が正しく特定されている。

以上より、提案したアルゴリズムでトランジスタレベルの故障追跡を正しく行な

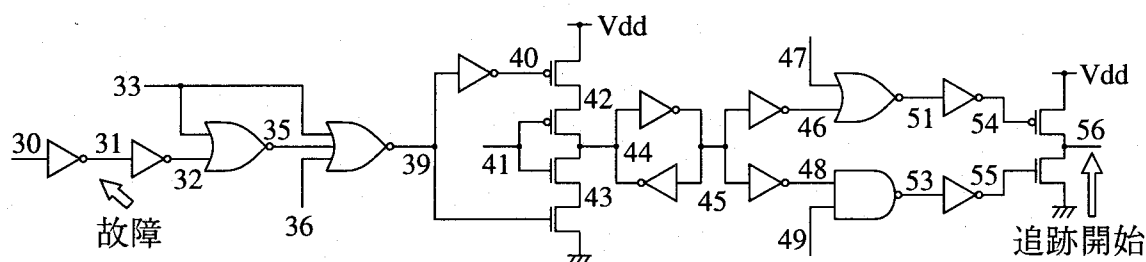


図 4.19 商用 DRAM の部分回路 2 のゲートレベル回路図

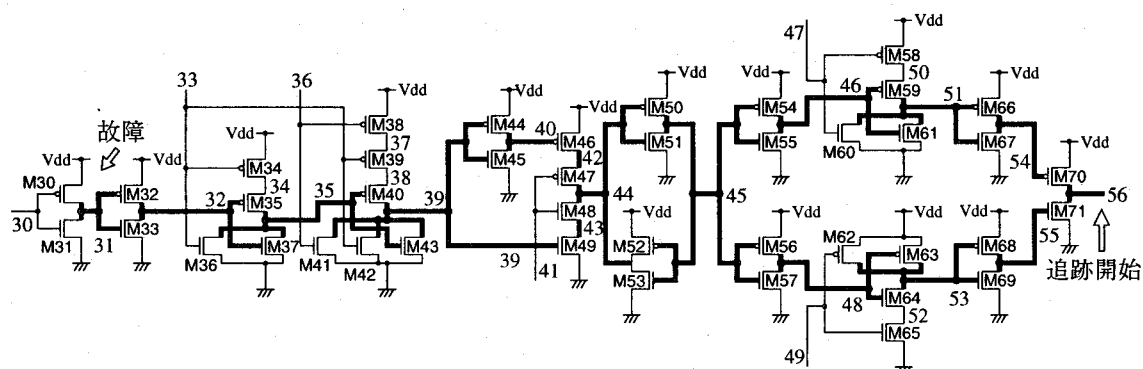


図 4.20 商用 DRAM の部分回路 2 のトランジスタレベル回路図

表 4.2 商用 DRAM の部分回路 2 に対するトランジスタレベル故障追跡の実行結果

1: measure[1]:55, Faulty	10: measure[10]:36, Good
2: measure[2]:53, Faulty	11: measure[11]:33, Good
3: measure[3]:48, Faulty	12: measure[12]:84, Faulty
4: measure[4]:45, Faulty	13: measure[13]:32, Faulty
5: measure[5]:44, Faulty	14: measure[14]:31, Faulty
6: measure[6]:41, Good	15: measure[15]:30, Good
7: measure[7]:42, Faulty	16: There may be a fault around the interconnection 31.
8: measure[8]:40, Faulty	
9: measure[9]:39, Faulty	

えることが確認できた。

4.8.2 平坦構造レイアウトからの逐次回路抽出による故障追跡

平坦構造レイアウトからの逐次回路抽出による故障追跡、および測定不能配線への対処法の有用性を示すため、実装したプログラムを図 4.21 に示す 4 ビット全加算器セルのレイアウトデータに対して適用した。このレイアウトは、ポリシリコン 1 層、金属 2 層で設計された CMOS 回路であり、平坦構造をしている。なお、故障追跡には関係のない基板コンタクト等は省略されている。あらかじめ想定した故障箇所および故障信号の伝搬経路をファイルに記録しておき、このファイルを参照して故障追跡を行なった。

図 4.22 は回路抽出により得られた本レイアウトのトランジスタレベルの回路図である。故障は配線 159 に存在すると仮定した。この配線は第 2 層金属を含まない測定不能配線である。ここから発生した故障信号が図 4.22 中で太線で示されている配線を伝搬していると想定して、この故障信号を配線 191 から追跡した。

プログラムの出力結果を表 4.3 に示す。表中の“measure”で始まる各行は、プロービング点の番号、配線番号、測定結果を示している。第 1 行は配線 191 が測定され、故障信号と判定されたことを示している。第 3 行は配線 203 が測定され、正常信号

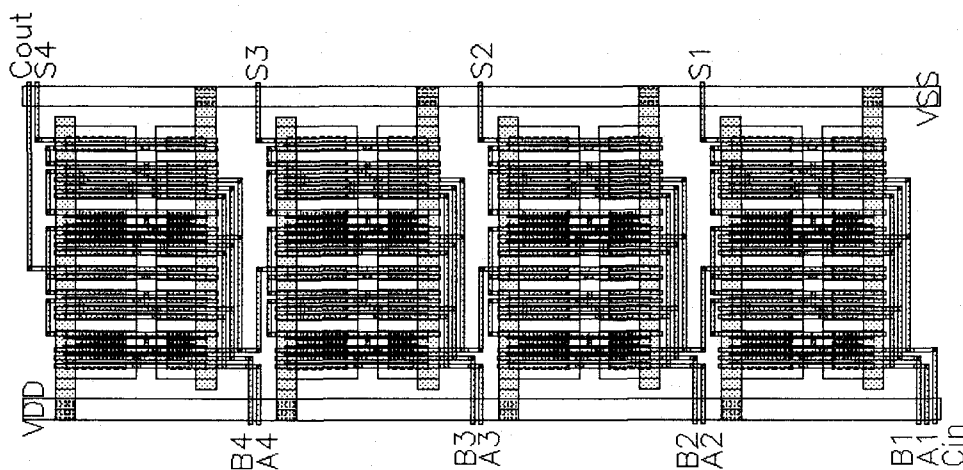


図 4.21 4 ビット加算器の平坦構造レイアウト

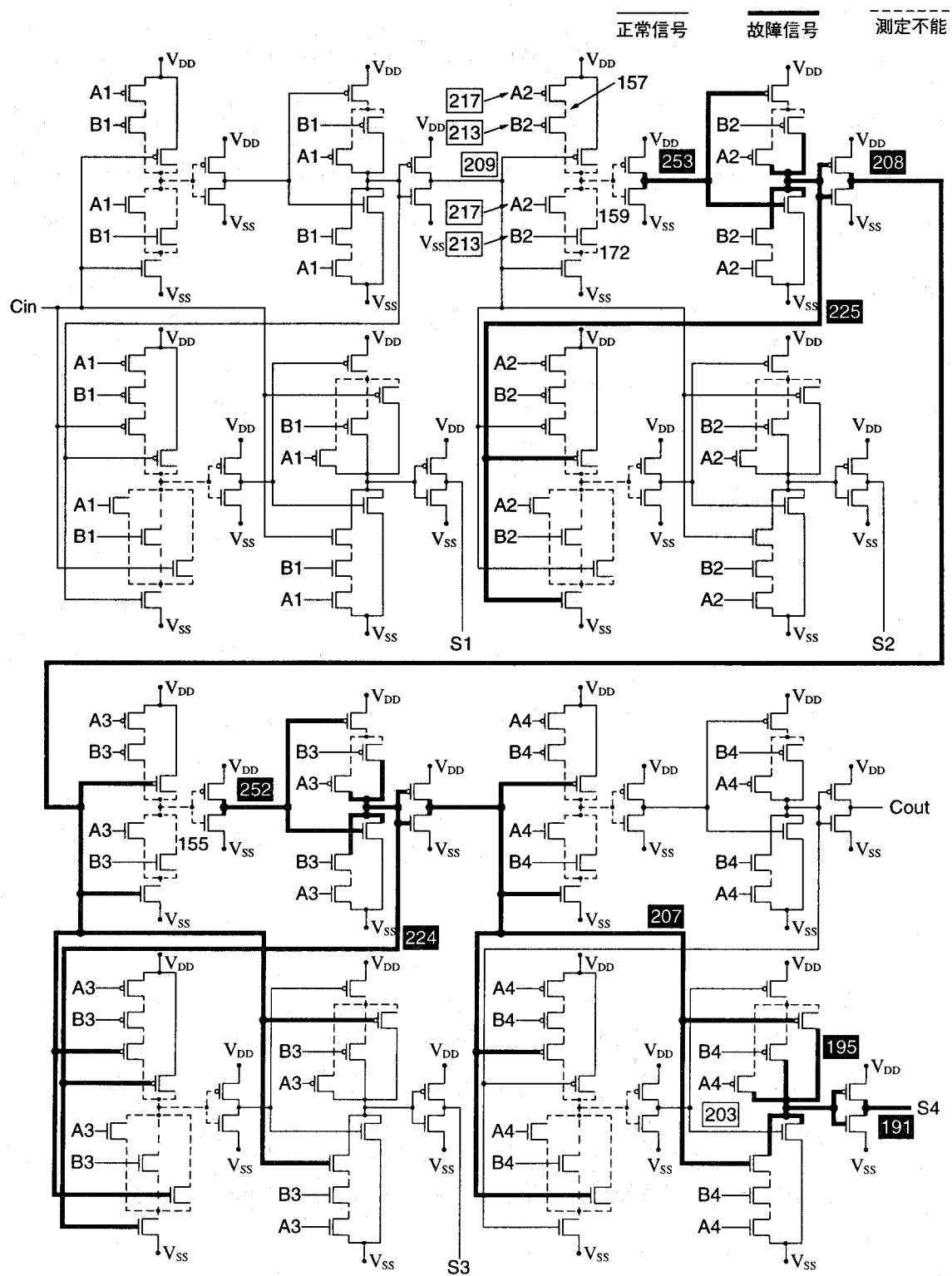


図 4.22 図 4.21 のレイアウトから抽出された回路図

表 4.3 4 ビット加算器の平坦構造レイアウトからの回路抽出・故障追跡の実行結果

1: measure[1]: 191, faulty	21: MOS-FET(2803)
2: measure[2]: 195, faulty	22: MOS-FET(2795)
3: measure[3]: 203, good	23: MOS-FET(2794)
4: measure[4]: 207, faulty	24: MOS-FET(2786)
5: measure[5]: 224, faulty	25: interconnection(157)
6: measure[6]: 252, faulty	26: MOS-FET(2787)
7: unmeasurable : 155	27: interconnection(172)
8: measure[7]: 208, faulty	28: MOS-FET(2802)
9: measure[8]: 225, faulty	29: MOS-FET(2810)
10: measure[9]: 253, faulty	30: Use an FIB after this.
11: unmeasurable : 159	31: measure[13]: 159, faulty
12: measure[10]: 209, good	32: measure[14]: 157, good
13: measure[11]: 213, good	33: measure[15]: 172, good
14: measure[12]: 217, good	34: There may be a fault around the interconnection 159.
15: unmeasurable : 157	35: interconnection(159)
16: unmeasurable : 172	36: MOS-FET(2803)
17: There may be a fault around the interconnection 253.	37: MOS-FET(2795)
18: interconnection(253)	38: MOS-FET(2794)
19: MOS-FET(2811)	39: MOS-FET(2786)
20: interconnection(159)	

と判定されたことを示している。第 7 行は配線 155 が測定不能配線であり測定が行われなかったことを示している。第一段階の追跡で絞り込まれた故障箇所が第 17 行から第 29 行に示されている。これらの結果は、図 4.22 および図 4.23 にも示されている。図中で四角で囲まれた数字は正常信号が測定された配線、白抜きの数字は故障信号が測定された配線、およびその他の数字は測定不能配線の番号である。図 4.23 のイタリック体の数字は MOS トランジスタの番号を表し、網かけされた領域が絞り込まれた故障箇所を示している。配線 253 とその上流に隣接する測定不能配線、およびそれらの上流 MOS トランジスタが指摘されている。

表 4.3 の第 30 行以降は、FIB を併用した第二段階の追跡である。第 34 行以降に絞り込み結果が示されている。配線 159 か、その上流の MOS トランジスタ 2803, 2795, 2794, 2786 に故障が存在することが指摘されている。この結果は、図 4.24 にも示されている。図 4.23 と同様に、網かけされた領域が絞り込まれた故障箇所である。これらの結果より、想定した故障が正しく同定されていることが分かる。

以上より、提案した手法およびアルゴリズムで、平坦構造レイアウトからの逐次

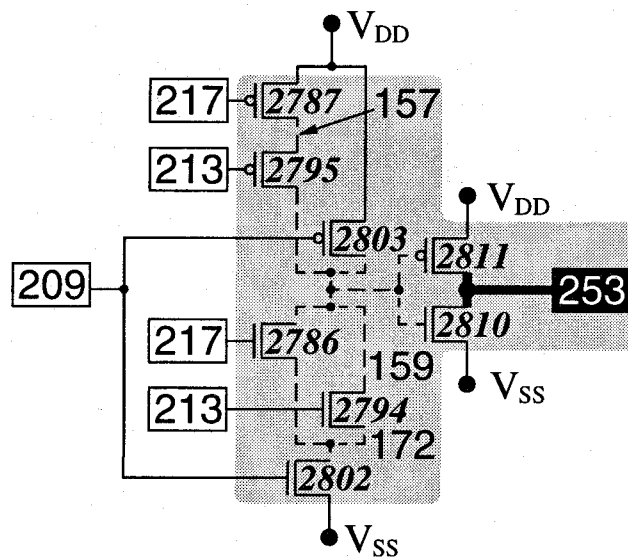


図 4.23 EB テスタのみを用いた第一段階の故障指摘箇所

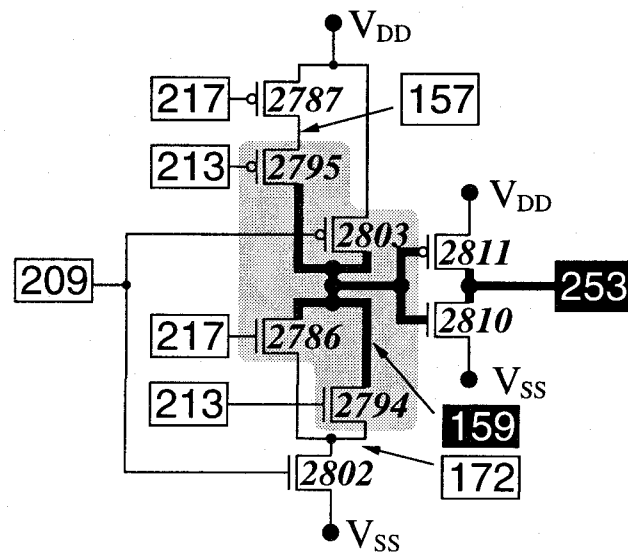


図 4.24 FIB を併用した第二段階の故障指摘箇所

回路抽出による故障追跡および測定不能配線への対処が正しく行なえることが確認できた。

4.8.3 階層構造レイアウトからの逐次回路抽出による故障追跡

階層構造レイアウトデータからの逐次回路抽出による故障追跡の有用性を示すため、実装したプログラムを図 4.25 に示す 16 ビット全加算器の階層構造レイアウトに対して適用した。このレイアウトは、2 層金属 CMOS プロセスで設計された CMOS 回路であり、約 600 トランジスタを含んでいる。基板コンタクトなどの故障追跡に直接関係のない要素は省略されている。あらかじめ想定した故障箇所および故障信号伝搬経路をファイルに設定しておき、このファイルを参照して故障追跡を行った。

図 4.26～4.28 は、図 4.25 から抽出された回路図である。故障は図 4.28 の配線 308 にあると仮定し、故障信号が太線で示した配線を伝搬していると設定した。この故障を図 4.26 の出力 S15 から追跡を行った。プログラムの出力結果をを表 4.4 に示す。表中の“measure”で始まる各行は、プロービング点の番号、配線番号、測定結果を示している。

表 4.4 の第 1～3 行は、図 4.26 の配線 78, 93 で故障信号が測定され、配線 108 で

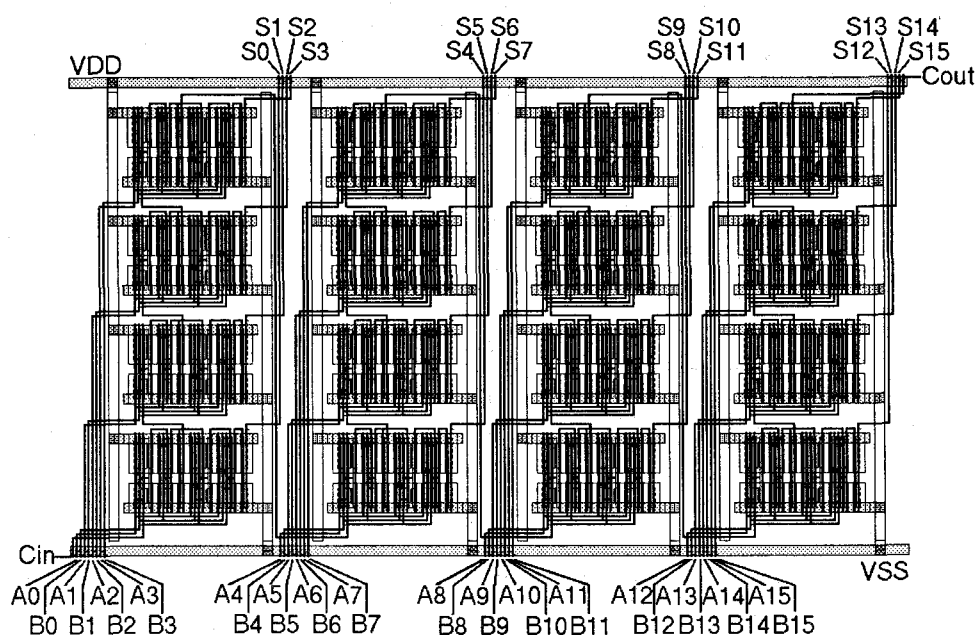


図 4.25 階層構造 16 ビット全加算器のレイアウト

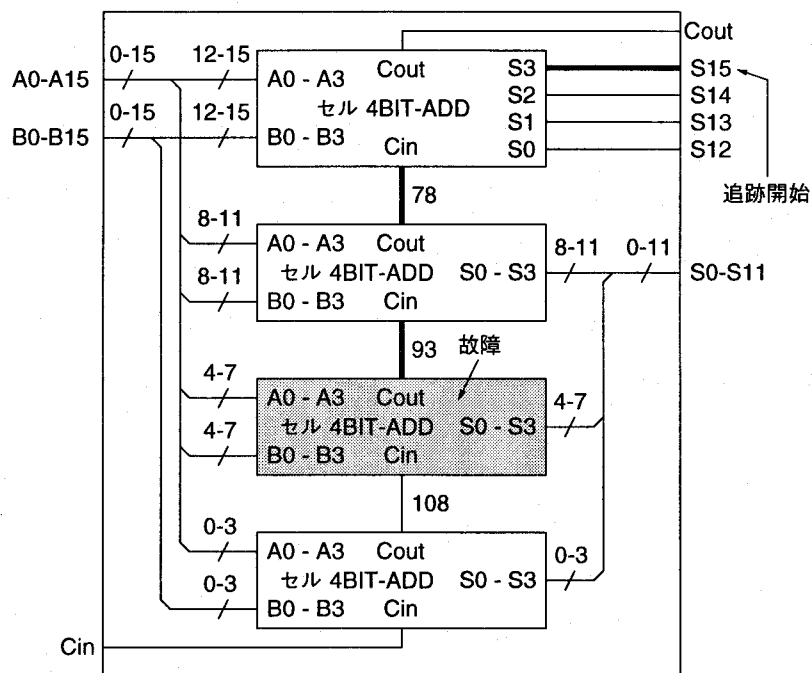


図 4.26 図 4.25 のレイアウトから抽出されたトップレベルの回路図

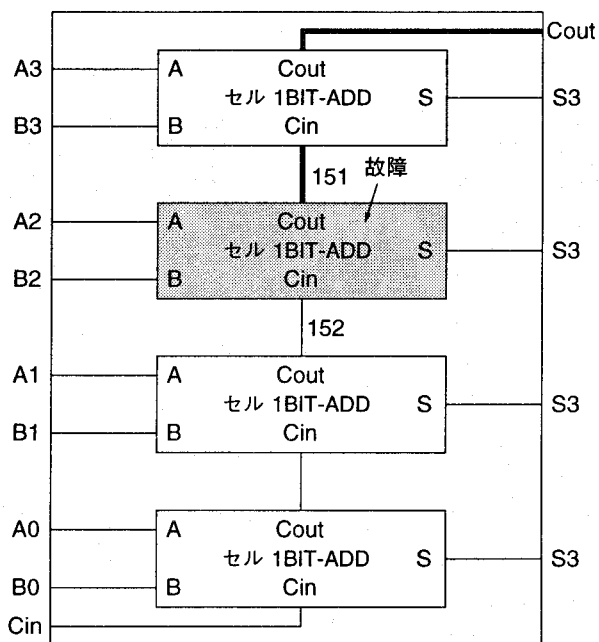


図 4.27 図 4.25 のレイアウトから抽出された第2レベルの4ビット加算器 (セル 4BIT-ADD) の回路図。

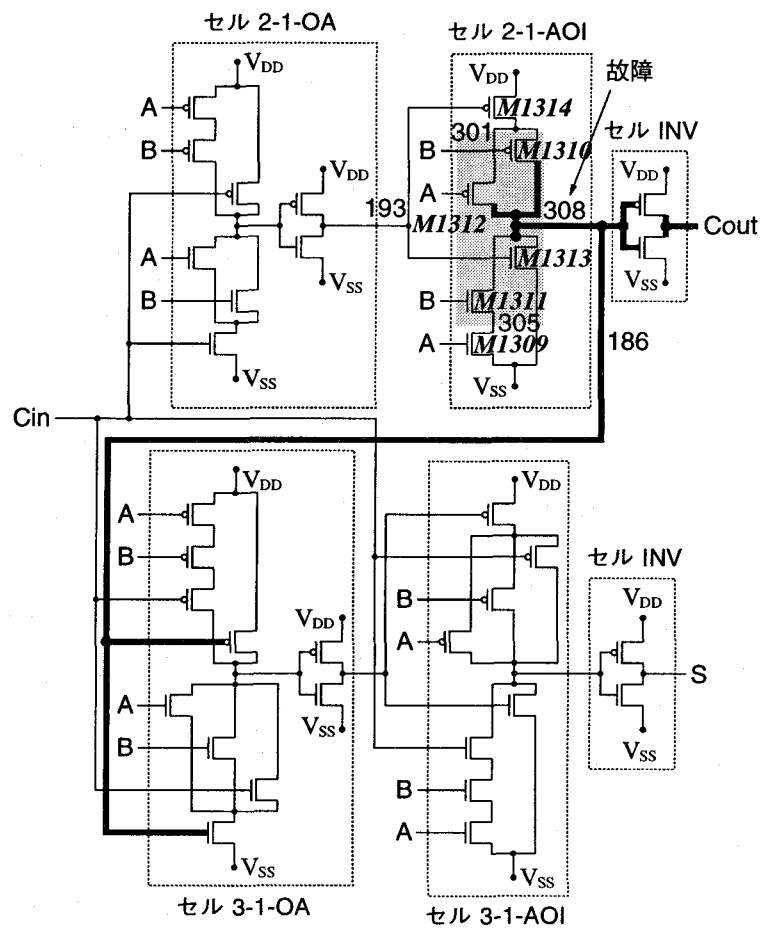


図 4.28 図 4.25 のレイアウトから抽出された第 3 レベルの 1 ビット加算器 (セル 1BIT-ADD) の回路図。

表 4.4 階層構造 16 ビット加算器の故障追跡結果

1: measure[1]:78, faulty	9: measure[9]:301, good
2: measure[2]:93, faulty	10: There may be a fault around the interconnection 308.
3: measure[3]:108, good	11: interconnection(308)
4: measure[4]:151, faulty	12: MOS-FET(1313)
5: measure[5]:152, good	13: MOS-FET(1311)
6: measure[6]:186, faulty	14: MOS-FET(1312)
7: measure[7]:193, good	15: MOS-FET(1310)
8: measure[8]:305, good	

正常信号が測定されたことを示している。これによって、3番目の網掛けされたセル 4BIT-ADD 内に故障があることが特定される。ここで、入力 A4-15, B4-15 が測定されていない。これは、これらの配線は主入力であり、これらの配線から電源やセルの出力端子に至るパスがないので、DC パスラベルが付けられないためである。

表 4.4 の第 4, 5 行および、第 6, 7 行はセル 4BIT-ADD およびセル 1BIT-ADD 内での故障追跡を示し、図 4.27 の 2 番目のセル 1BIT-ADD、図 4.28 のセル 2-1-AOI の内部に故障があることが特定される。第 8, 9 行は、セル 2-1-AOI 内の配線 305, 301 の両方で正常信号が測定されたことを示し、この結果、第 10 行以降で配線 308 か MOS トランジスタ M1313, M1311, M1212, M1210 のいずれかに故障があることが示されている。以上のように、想定された故障が正しく指摘された。

階層構造レイアウトからの逐次回路抽出、階層適故障追跡の効率を評価するため、図 4.25 のレイアウトを平坦構造化して故障追跡し、階層構造レイアウトの場合と CPU 時間、メモリ使用量、プロービング点数の比較を行った。この結果を表 4.5 に示す。表より階層構造レイアウトデータを用い、階層的に故障追跡することで CPU 時間が 38 倍、メモリ使用量が 7 倍、プロービング点数が 3.7 倍改善されていることが分かる

表 4.5 平坦構造レイアウトの場合と階層構造レイアウトの場合の故障追跡効率の比較

	CPU 時間 [sec]	メモリ使用量 [M bytes]	プロービング点数
平坦構造レイアウト	58.6	7.28	33
階層構造レイアウト	1.52	1.04	9

以上より、提案した手法およびアルゴリズムで階層構造レイアウトデータからの逐次回路抽出手による故障追跡を正しく行なうことができ、また階層構造化により処理効率が大きく改善されることが確認できた。

4.8.4 双方向バス配線を含む順序回路の故障追跡

順序回路、双方向バス配線に対する故障追跡アルゴリズムの有効性を示すため、

実装したプログラムを図 4.29 に示すモデル 8 ビットマイクロプロセッサのレイアウトデータに適用した。図中の ACC はアキュムレータ、ALU は算術論理演算ユニット、I/O は入出力をそれぞれ意味する。このレイアウトデータは、金属 2 層、ポリシリコン 1 層の CMOS 技術を用いて設計されており、約 2600 のトランジスタを含んでいる。あらかじめ想定した故障箇所、故障信号伝搬経路をファイルに記録しておき、このファイルを参照して追跡を行った。図 4.29 のレイアウトから抽出された回路図を図 4.30 に示す。故障はセル ALU 内の網かけされたセル ALU-1BIT 内にあり、太線で示された配線を故障信号が伝搬していると想定した。配線 45 に到達した故障信号は、一旦、セル REGFILE 内の記憶素子に保存され、異なる位相において再び配線 45 を通過して、外部出力端子に接続されている配線 5 へと伝搬していると想定した。

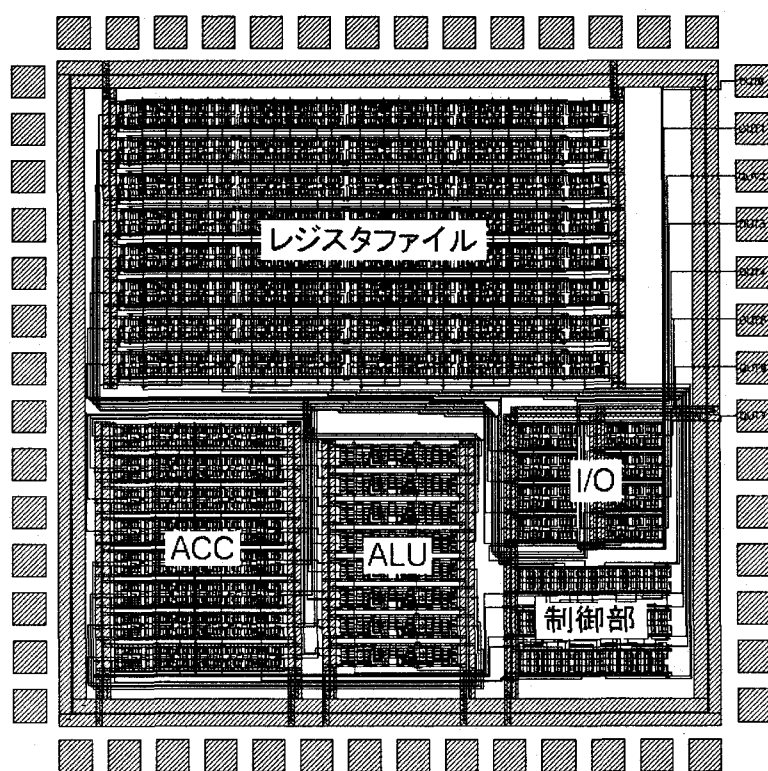


図 4.29 8 ビット CMOS マイクロプロセッサのレイアウト

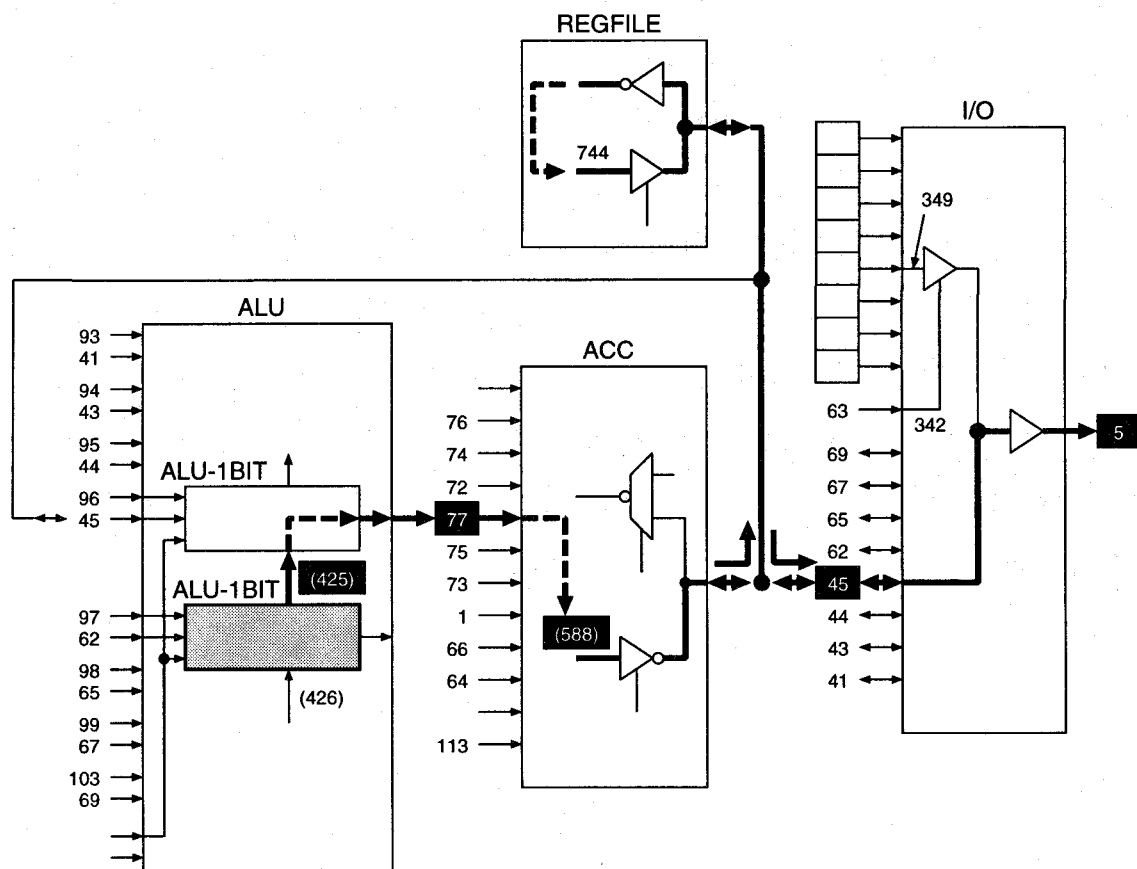


図 4.30 図 4.29 のレイアウトから抽出された回路図

プログラムの出力結果を表 4.6 に示す。表中の“measure”で始まる各行は、プロービング点数、配線番号、測定位相範囲、波形比較結果、および故障信号の場合は故障信号の位相範囲を示している。ここで、位相の単位は ns である。表中の第 3～10 行で、配線 41, 43, 44, 45 が双方向バス配線と判定され、これらの配線が測定されている。これらの配線については、測定開始位相が 0ns になっている。このうち、配線 45 で故障信号が測定されている。配線 45 では、二度故障信号が測定されるが、出力結果に表示されているのは、このうち位相の早い方のみである。このように、双方向バス配線を伝搬する故障信号も効率良く追跡され、セル ALU-1BIT 内に想定した故障箇所が、正しく指摘された。

表 4.6 モデル 8 ビットマイクロプロセッサの故障追跡結果

1: Initial Fault Phase: [805-825]
 2: measure[1]:63, Phase: [705-825], Good
 3: Interconnection 41 may be a bus.
 4: measure[2]:41, Phase: [0-825], Good
 5: Interconnection 43 may be a bus.
 6: measure[3]:43, Phase: [0-825], Good
 7: Interconnection 44 may be a bus.
 8: measure[4]:44, Phase: [0-825], Good
 9: Interconnection 45 may be a bus.
 10: measure[5]:45, Phase: [0-825], Faulty: [262-282]
 11: Cheking output driver of I/O
 12: measure[6]:349, Phase: [162-282], Good
 13: measure[7]:342, Phase: [162-282], Good
 14: Cheking output driver of REGFILE
 15: measure[8]:744, Phase: [162-282], Good
 16: Cheking output driver of ACC-8BIT
 17: measure[9]:588, Phase: [162-282], Faulty: [261-281]
 18: measure[10]:113, Phase: [162-282], Good
 19: measure[11]:66, Phase: [162-282], Good
 20: measure[12]:64, Phase: [162-282], Good
 21: measure[13]:71, Phase: [162-282], Good
 22: measure[14]:72, Phase: [162-282], Good
 23: measure[15]:73, Phase: [162-282], Good
 24: measure[16]:74, Phase: [162-282], Good
 25: measure[17]:75, Phase: [162-282], Good
 26: measure[18]:76, Phase: [162-282], Good
 27: measure[19]:77, Phase: [162-282], Faulty: [207-227]
 28: measure[20]:70, Phase: [107-227], Good
 29: measure[21]:68, Phase: [107-227], Good
 30: measure[22]:93, Phase: [107-227], Good
 31: measure[23]:94, Phase: [107-227], Good
 32: measure[24]:95, Phase: [107-227], Good
 33: measure[25]:96, Phase: [107-227], Good
 34: measure[26]:97, Phase: [107-227], Good
 35: measure[27]:99, Phase: [107-227], Good
 36: measure[28]:103, Phase: [107-227], Good
 37: measure[29]:98, Phase: [107-227], Good
 38: Interconnection 62 may be a bus.
 39: measure[30]:62, Phase: [0-227], Good
 40: Interconnection 65 may be a bus.
 41: measure[31]:65, Phase: [0-227], Good
 42: Interconnection 67 may be a bus.
 43: measure[32]:67, Phase: [0-227], Good
 44: Interconnection 69 may be a bus.
 45: measure[33]:69, Phase: [0-227], Good
 46: There is a fault in the cell: ALU-8BIT
 47: measure[34]:425, Phase: [107-227], Faulty: [205-225]
 48: measure[35]:426, Phase: [105-225], Good
 49: There is a fault in the cell: ALU-1BIT
 50: measure[36]:478, Phase: [105-225], Good
 51: There is a fault in the cell: ALU-SUM
 52: measure[37]:1153, Phase: [105-225], Good
 53: There may be a fault around the interconnection 1138
 54: interconnection(1138)
 55: MOS-FET(4806)
 56: MOS-FET(4792)

4.8.5 プロービング点数の評価

プロービング点数を評価するため、ISCAS'89 ベンチマーク回路情報からレイアウトデータを生成し、本手法を適用した。レイアウトは、CMOS3 セルライブラリー^[55]に含まれる基本セルを自動配置・配線ツールで配置・配線して生成した。ポリシリコン1層、金属3層のプロセスを利用している。各回路に対して、ランダムに10箇所の性能故障を仮定し、ここで発生した故障信号を主出力まで伝搬させて、故障追跡を行った。

例として、順序回路ベンチマーク回路 s27 の一つの故障について、本手法で故障追跡した結果を示す。図 4.31 および図 4.32 は回路 s27 のレイアウトおよびレイアウトから抽出したゲートレベル回路図である。また、図 4.33 は図 4.32 のゲート "L" のトランジスタレベル回路図である。ここで、トランジスタ M564 に故障があると想定し、この故障信号を図 4.32 の配線 86 から追跡した。プログラムの出力結果を表 4.7 に示す。表中の“measure”で始まる各行には、測定点の番号、配線番号、測定位相範囲、波形比較結果、および故障信号の場合は故障信号の位相範囲が示されている。なお、位相の単位は ns である。第 10 行目は指摘された故障ゲートを示し、第 13～15 行は最終的に絞りこまれたトランジスタレベルの故障範囲を示している。この範囲は、図 4.33 において網掛けされて表示されている。このように、仮定した故障箇所が正しく指摘されている。

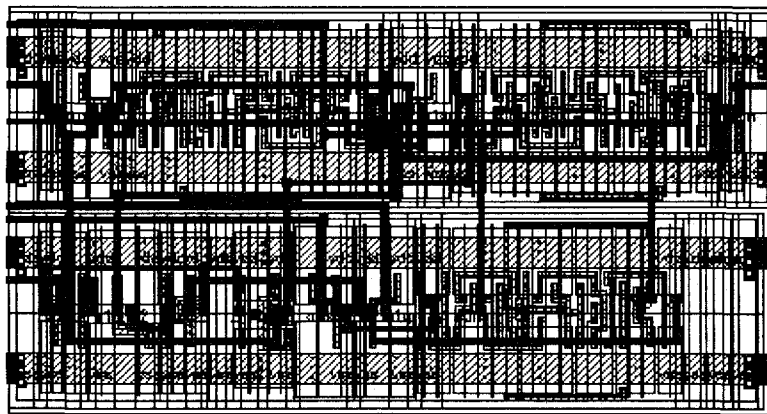


図 4.31 ISCAS'89 ベンチマーク回路情報 s27 から生成したレイアウト

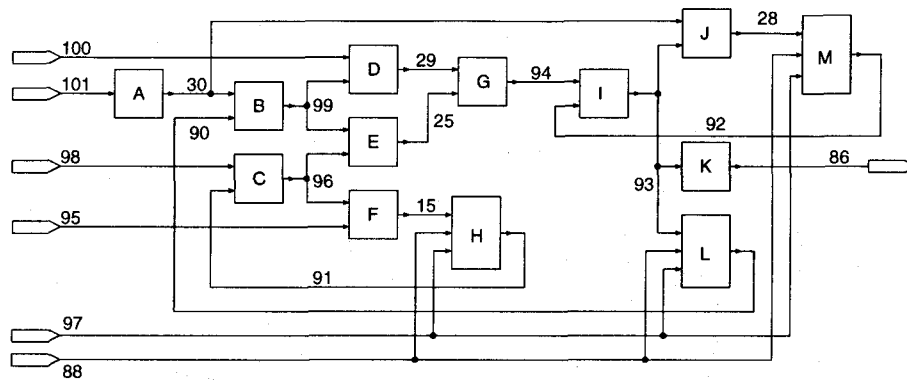


図 4.32 図 4.31 のレイアウトから抽出したゲートレベル回路図

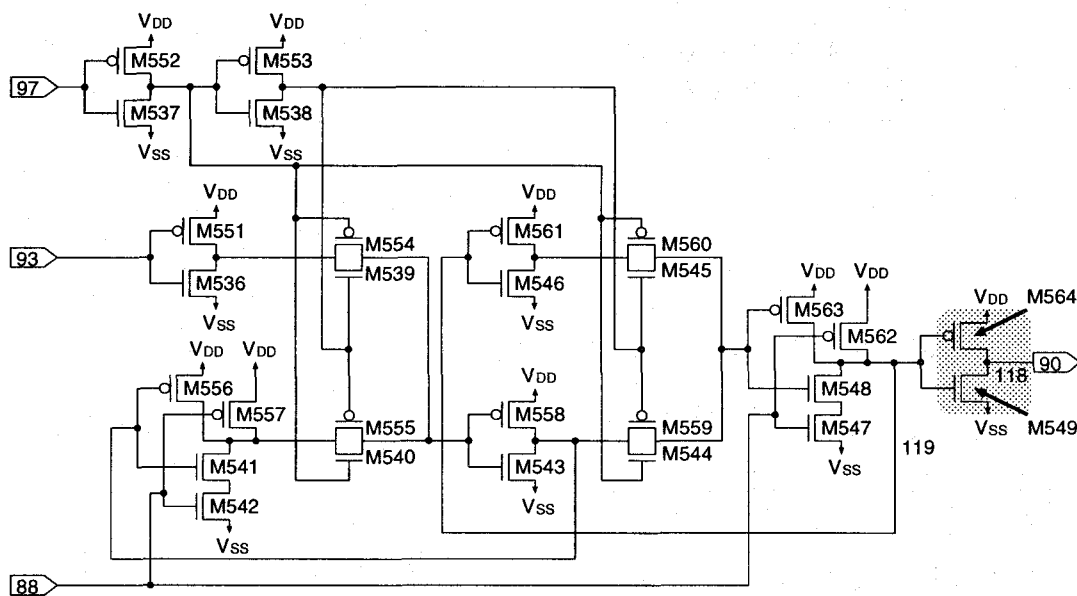


図 4.33 図 4.32 のゲート “L” のトランジスタレベル回路図。ここで、トランジスタ M562 が故障していると仮定されており、網掛けされた範囲が故障箇所として指摘された。

ISCAS'89 ベンチマーク回路の各回路に対して、ランダムに単一故障を仮定し、この故障信号を上述の s27 と同様に追跡する操作を 10 回行った。この時のプロービング点数の平均を表 4.8 に示す。表には、左の列から順に、回路名、トランジスタ数、10 回の追跡の平均プロービング点数が示されている。表より、回路の中に含まれる

表 4.7 プログラム実行結果

1: Initial Fault Phase: [210-230]
 2: measure[1]:93, Phase: [110-230], Faulty: [208-228]
 3: measure[2]:92, Phase: [108-228], Good
 4: measure[3]:94, Phase: [108-228], Faulty: [206-226]
 5: measure[4]:25, Phase: [106-226], Faulty: [204-224]
 6: measure[5]:99, Phase: [104-224], Faulty: [202-222]
 7: measure[6]:90, Phase: [102-222], Faulty: [200-220]
 8: measure[7]:88, Phase: [100-220], Good
 9: measure[8]:97, Phase: [100-220], Good
 10: There is a fault in the cell: L
 11: measure[9]:119, Phase: [100-220], Good
 12: There may be a fault around the interconnection 118
 13: interconnection(118)
 14: MOS-FET(564)
 15: MOS-FET(549)

表 4.8 本手法とガイドッドプローブ法のプロービング点数の比較

回路名	トランジスタ数	平均プロービング点数	回路名	トランジスタ数	平均プロービング点数
S27	136	6.6	S713	1978	9.6
S208	676	14.9	S820	1940	16.2
S298	1006	10.9	S832	1976	17.6
S344	1098	11.7	S838	2860	15.2
S349	1108	9.7	S953	2530	9.0
S382	1316	29.5	S1196	3000	10.7
S386	1114	10.7	S1238	3118	9.1
S400	1350	16.0	S1423	5220	18.1
S420	1404	12.7	S1488	4058	12.3
S444	1392	13.4	S1494	4086	16.9
S510	1158	10.9	S5378	14276	17.2
S526	1692	15.3	S9234	25048	30.0
S641	1858	11.7	S13207	44726	23.1

トランジスタ数が大きく変わっても、プロービング点数はあまり変わらず、実用的なプロービング点数で故障追跡が完了していることが分かる。以上のことより、本手法は CAD データからの逐次回路抽出および EB テスタによる波形測定を実用的な時間内に実行することができ、VLSI の故障追跡に有効であることが確認できた。

4.9 むすび

本章では、階層構造レイアウトデータから抽出した回路データの利用を前提とした、階層的故障追跡アルゴリズムについて述べた。本アルゴリズムの特徴は以下の通りである。

- (1) 信号の流れの向きに関する情報の含まれていないレイアウトデータからの抽出回路を解析し、信号の流れの向きを判定しながら追跡を行う。
- (2) 電圧測定精度の低下する絶縁膜下の配線を回避して追跡可能である。
- (3) 絶縁膜への穴空け、パッド作成が可能な FIB 加工装置が利用可能であれば、EB テスタと併用して、効率良く故障箇所を絞り込むことができる。
- (4) テストシーケンスが長い順序回路においては、故障信号が測定される位相範囲のみを測定し、故障追跡に伴って測定位相を制御することによって、測定時間の短縮が可能である。
- (5) 順序回路を含んでいる場合も追跡可能である。
- (6) 双方向バス配線を含んでいる場合も追跡可能である。

第5章 自動故障追跡システムの構築^[56]

5.1 まえがき

本章では、第3章、第4章の成果をふまえ、CADレイアウトからの逐次回路抽出法および階層的故障追跡アルゴリズムを実装した、EB自動故障追跡システムの構築について述べる。まず、本システムのハードウェア構成、ソフトウェア構成について述べる。つぎに、本システムを商用のCMOS画像処理プロセッサの故障に対して適用して、本システムがVLSIの自動故障追跡に有効であることを示す。

5.2 ハードウェア構成

自動故障追跡システムのハードウェア構成を図5.1に示す。システムは、CADリンクEBテストシステムとLSIテストシステムとを接続して構築したもので、以下の装置からなる。

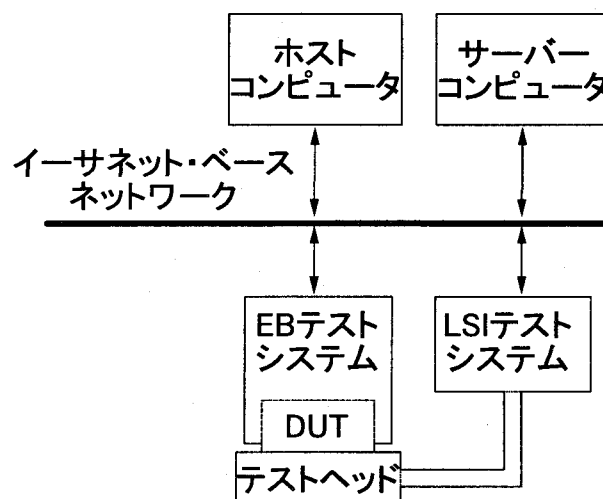


図 5.1 自動故障追跡システムのハードウェア構成

- (1) EB テストシステム (Schlumberger Technologies 社製 IDS 5000ZX)
- (2) LSI テストシステム (Hewlett Packard 社製 HP83000)
- (3) サーバー計算機 (Hewlett Packard 社製 HP9000/J210: 168SPECint92, 269SPECfp92)
- (4) ホスト計算機 (SiliconGraphics 社製 Indigo²: 119SPECint92, 131SPECfp92)

これらの装置は、イーサネットベースのネットワークにより接続されている。

EB テストシステムは、電子銃室が下、試料室が上という、通常の SEM とは上下逆の配置になっており、DUT のピン接続をテーブル上面で行える。LSI テストシステムのテストヘッドと DUT ロードモジュールを直結するダイレクト・ドッキングを行うことにより、ピン数の多い DUT でも高速動作状態でテストが可能である。

LSI テストシステムは、最大で 176 ピン、最高動作周波数 120MHz の VLSI を駆動させることができる。

サーバーコンピュータは 100 万トランジスタクラスの VLSI のレイアウトデータを 10 チップ程度格納することができる。

ホストコンピュータは、100 万トランジスタ程度の階層構造レイアウトデータを読み込んで故障追跡を行うことができる。

5.3 ソフトウェア構成

以下の 6 つのプログラムを統合して、故障追跡ソフトウェアを実装した。

- (1) 故障追跡プログラム：故障追跡の制御、他のプログラムの呼び出しを行う。
- (2) 波形測定の為の最適プロービング点選択プログラム：EB テスタによる波形測定に適したプロービング点を選択する。
- (3) DUT 配線パターンと CAD レイアウトのマッチングプログラム：EB テスタの試料ステージ移動誤差を補正する為、DUT 配線パターンと CAD レイアウトのマッチング位置を求める。

- (4) 波形比較プログラム：正常信号と故障信号の判定を行うため、DUT の測定波形と良品の測定波形から波形パラメータを抽出し、比較する。
- (5) EB テスタの制御プログラム：EB テスタの各種ハードウェアを制御し、試料ステージの移動、SEM 像の取得、波形の取得等を行う。
- (6) LSI タスタの制御プログラム：LSI テスタの各種ハードウェアを制御し、DUT に与えるテストシーケンスの設定を行う。

以降の節で、これらのプログラムに付いて順次説明する。

5.3.1 故障追跡

故障追跡プログラムに CAD レイアウトデータ、故障追跡を開始する配線、および故障信号の位相を与えると、プログラムはレイアウトから回路データを抽出し、測定を行う上流の測定可能配線を選択して出力する。これに対して、この配線の測定結果を入力すると、プログラムはその結果に応じて次の測定配線を選択し、出力する。故障箇所の推定もこのプログラムが行う。

また、本プログラムが、システムのメインプログラムとなっている。

5.3.2 最適プロービング点選択

最適プロービング点選択プログラム^[57]は、故障追跡プログラムから以下のデータを受け取る。

- (1) 測定を行う上流配線の周辺に存在する配線構成ポリゴンのデータ
- (2) 信号の流れの向きに関するデータ

最適プロービング点選択プログラムは、第 2.4.3 節で述べた以下のルールに従って、プロービング点の選択を行う。

- (1) なるべく配線の太い部分を選択する。

- (2) できる限り上層の金属部分を選択する。
- (3) コンタクト領域は避ける。
- (4) 局所電界効果が測定に悪影響を及ぼす恐れの高い領域を避ける。
- (5) 配線の中で、なるべく信号の下流部分 (次段の素子の入力に近い部分) を選択する。
- (6) 他の上層金属配線によって隠れていない部分を選択する。

プログラムは、選択されたプロービング点の X-Y 座標を出力する。

5.3.3 パターンマッチング

SEM 像と CAD レイアウト像のパターンマッチングには、ハイブリッド GA (Genetic Algorithm, 遺伝的アルゴリズム) による方法^[58]を用いている。プログラムは、マッチング位置での SEM 画像の左上角の座標、CAD レイアウト像に対する SEM 画像の倍率、回転角度を出力する。この出力値を用いて、メインプログラムは電子ビームのプロービング点の位置補正を行う。

5.3.4 波形比較

波形比較には、波形のパラメータを抽出して比較する手法^[59]を用いている。性能故障の追跡では、波形の時間的パラメータが問題となり、正確かつ高速な波形比較が必要だからである。波形の振幅、立ち上がりの時間・時刻、たち下がりの時間・時刻がパラメータとして抽出・比較される。

波形パラメータの抽出法としては、関数近似による方法 (FAM 法) とノイズ除去フィルタによる方法 (FIL 法) とを選択することができる。FAM 法はノイズに強いが処理時間がかかるという特徴がある。逆に、FIL 法は高速だがノイズの影響を受けやすい。プログラムは、DUT の波形と良品の波形から抽出したパラメータを比較し、正常波形である (二つの波形が同一である) か、故障波形 (二つの波形が異なる) かを判定して、その結果を出力する。故障波形の場合には、故障の位相も出力する。

比較が困難な場合は、その旨出力する。その場合、メインプログラムはその配線を測定不能配線と同等に扱う。

5.3.5 EB テスタの制御

EB テストシステムの制御は、IDS Command Line Application Programming Interface (Schlumberger Technologies 社製) を通じて行う。

5.3.6 LSI テスタの制御

LSI テストシステムの制御は、C 言語で書かれた LSI テスタ制御関数ライブラリ (Hewlett Packard 社製) を通じて行う。

5.3.7 各プログラムの関係

ソフトウェアの構成を図 5.2 に示す。プログラム (2)～(6) は、ローカル・プロセス呼出しによってメインプログラムから呼び出される。さらに、プログラム (5)、(6) は、リモート・プロセス呼び出しによって、それぞれ EB テストシステム、LSI テストシステムの制御プログラムを呼び出す。プロセス間のデータ通信については、UNIX オペレーティングシステムのパイプ機能およびファイルを介して行われる。

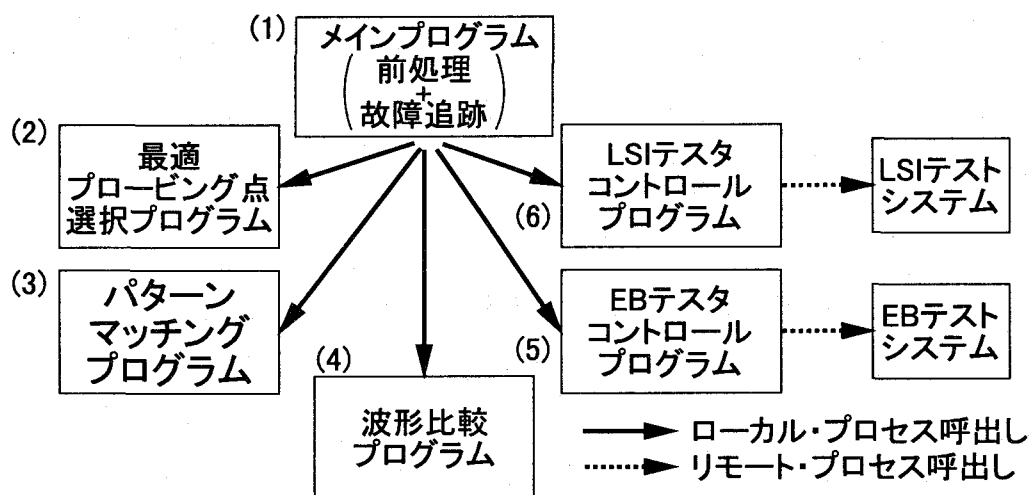


図 5.2 自動故障追跡システムのソフトウェア構成

5.4 システム性能

本システムの性能を表 5.1 に示す。ここで、SEM 画像のピクセル数は 512×512 で、256 階調の階調画像である。メインプログラムの実行時間の大部分は、CAD レイアウトデータの前処理に費されており、この時間は CAD レイアウトデータの規模に依存する。この例では、レイアウトデータのサイズは 130kB である。メインプログラムは、1 つの DUT で 1 回の故障追跡に対して 1 回呼び出される。これに対して、その他のプログラムは波形測定回数分だけ呼び出される。従って、故障追跡時間の合計のうちで支配的な時間は、波形の測定時間である。

表 5.1 システム性能

プログラム	内容	実行時間(1 回あたり)[sec]
メイン	前処理+故障追跡	~140 (CAD レイアウトデータのサイズに依存)
最適プロービング点選択		10
DUT 配線パターンと CAD レイアウトとのパターンマッチング		14
波形比較	FAM	17
	FIL	0.3
EB テスタの制御	X-Y ステージの移動	5
	SEM 像の取得	10
	波形の取得	30
LSI テスタの制御		≪ 1

5.5 応用

故障追跡システムの有用性を示す為、本システムを商用の CMOS 画像処理プロセッサ (トランジスタ数約 54 万個) の回路ブロック (トランジスタ数約 4700 個) に適用した。図 5.3 に、この回路ブロックのレイアウト図を示す。この回路ブロックは 0.8 μ m の 2 層金属、1 層ポリシリコン CMOS プロセスで設計・製造されている。

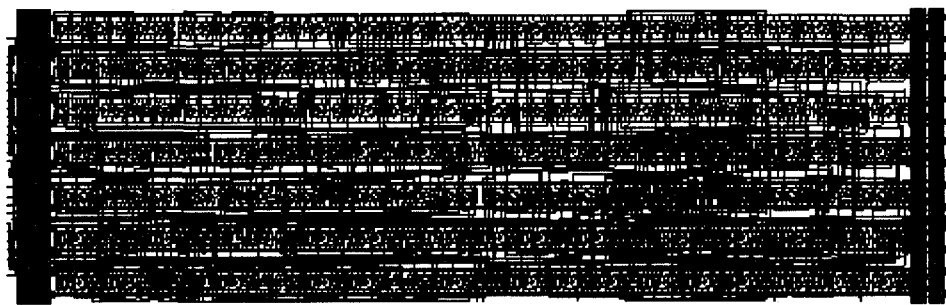


図 5.3 商用 CMOS 画像処理プロセッサの回路ブロックのレイアウト

この回路ブロックの入力配線 973 に与える信号を、図 5.4 の実線から破線に変化させると、出力配線 265 の位相 770ns から 970ns で故障信号が測定された。このように、正常状態と故障状態は LSI テスタコントロールプログラムで入力信号を変化させることで作り出した。

本システムを用いてこの故障を出力配線 265 から追跡したところ、以下のように追跡が進められた。

- (1) まず始めに、システムからオペレータに対して電源やグランド等の定電圧配線の信号を測定するよう指示が出される。これは、波形比較プログラム^[59]において雑音の標準偏差 σ が必要となるためである。指示に従って、手動で EB テスタを操作し、定電圧の配線の波形を測定する。
- (2) 次に、システムからオペレータに対して、CAD レイアウトと SEM 像の位置

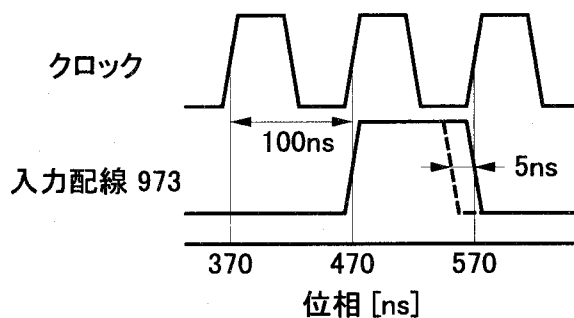


図 5.4 CMOS 画像処理プロセッサに与える入力信号波形

合わせをするよう指示が出される。指示に従って、チップの角のうち3点を用いて位置合わせを行う。

- (3) LSI テスタのみによるテストで故障が見つかった出力配線 (配線 265)、故障信号の位相 [770ns, 970ns] を入力すると、故障追跡が開始される。
- (4) メインプログラムにより必要な部分の回路を抽出し、測定配線を決定する。決定された測定配線の周辺のレイアウトデータがファイルに記録され、プロービング点決定プログラムが起動される。この様子が図 5.5 に示されている。また、レイアウトデータとともに、信号の流れの向きに関する情報も渡されている。
- (5) プローブ点決定プログラムによって、最適なプロービング点が決定され、その座標情報がメインプログラムに渡される。
- (6) EB テスタコントロールプログラムが呼び出され、プロービング点が画面の中心となるように、EB テスタのステージが移動される。
- (7) EB テスタコントロールプログラムによって配線幅が 10 ピクセル程度になるように倍率調整が行われ、SEM 像が取得される。
- (8) 画像マッチングプログラムが呼び出され、SEM 画像を 2 値化し (配線領域を

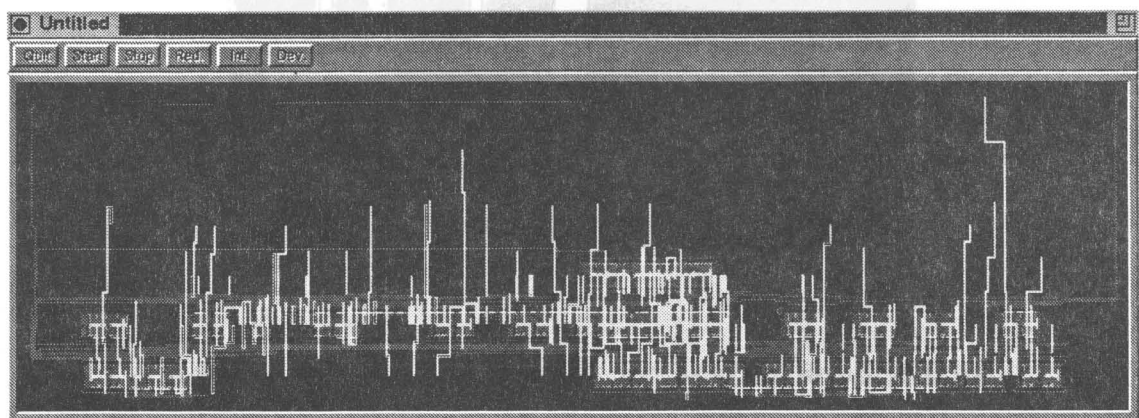


図 5.5 最適プロービング点決定プログラムのウィンドウ

白に、その他の領域を黒にする)、CAD レイアウト画像とのマッチングが行われる。例を図 5.6 に示す。(a) が SEM 画像、(b) が二値化後の画像である。二値化画像と CAD レイアウト画像とのマッチングにより、EB プロービング位置の誤差が算出される。マッチング座標において CAD 画像と SEM 画像を重ね合わせた例を図 5.7 に示す。

- (9) EB テスタコントロールプログラムにより、プロービング点の位置が補正され、最適プロービング点において、位相 [370ns, 870ns] の波形 (A) が取得され、ファイルに記録される。測定位相は、メインプログラムにより指示されたものである。
- (10) LSI テスタコントロールプログラムが呼び出され、正常出力を与える入力信号と故障出力を与える入力信号を切り替える。
- (11) EB テスタコントロールプログラムが呼び出されて、同じ点、同じ位相で波形 (B) が取得され、ファイルに記録される。
- (12) 測定された二つの波形 (A)、(B) が波形比較プログラムにより比較される。
- (13) 手続き (4)~(12) が、故障箇所が同定されるまで繰り返される。

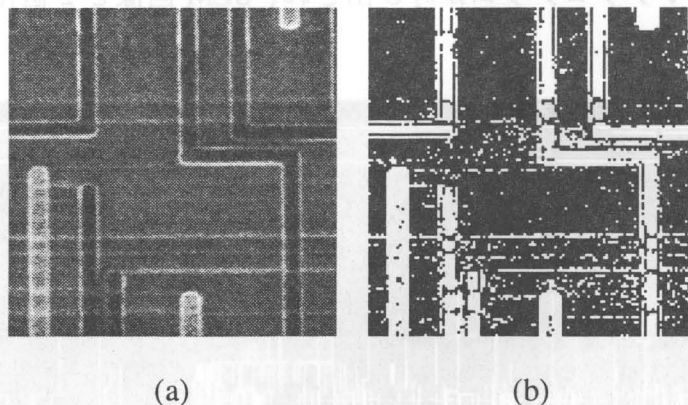


図 5.6 (a)SEM 画像の例, (b)SEM 画像を配線領域が白色、他が黒色となるように二値化した画像

メインプログラムの出力結果を表 5.2 に示す。表中の“measure”で始まる各行には、測定点の番号、配線番号、測定位相範囲、波形比較結果、および故障信号の場合は故障信号の位相範囲が示されている。なお、位相の単位は ns である。

第 2 行～12 行ではセルレベルの故障追跡の結果を示している。この追跡経路のセルレベル回路図を図 5.8 に示す。ここで、セル FF はフリップフロップであり、D はデータ入力、RST はリセット入力、CLKA、CLKB は相補的なクロック入力、Q はデータ出力を表している。また、図中で追跡経路は太線で示されており、網掛けされて示されているセル FF が故障セルであると指摘されている。続いて故障セル FF 内がトランジスタレベルで追跡されている。表 5.2 の第 14～22 行にトランジスタレベル故障追跡の結果が示されている。セル FF 内のトランジスタレベル回路図を図 5.9 に示す。表 5.2 の第 23～26 行に故障の可能性のある配線と MOS トランジスタの名前が示されている。指摘された配線および MOS トランジスタは故障信号が入力されている配線 973 に接続されている配線および MOS トランジスタである。我々の故障追跡アルゴリズムでは、外部入力配線は常に正しいと仮定しているので、こ

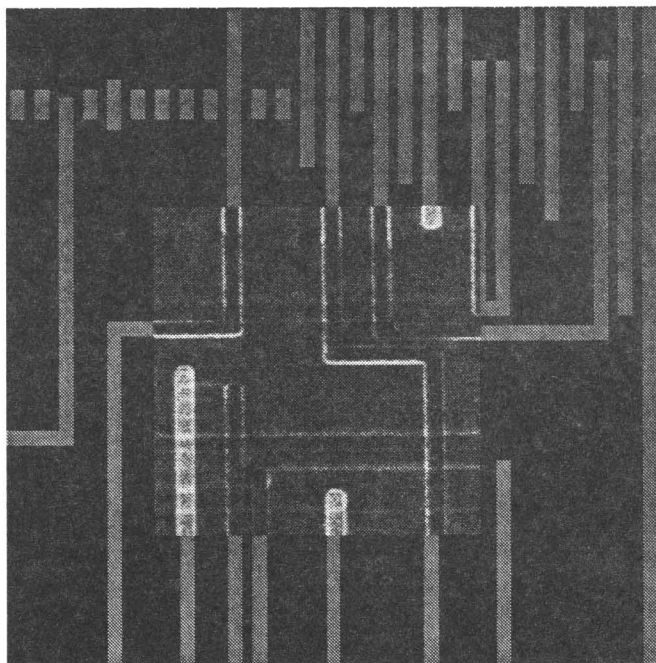


図 5.7 SEM 画像と CAD レイアウト画像のマッチングの例

表 5.2 自動故障追跡システム出力結果

1: Initial fault phase: [770, 970]
 2: measure[1]:54, Phase[370, 870], Good
 3: measure[2]:62, Phase[370, 870], Good
 4: measure[3]:377, Phase[370, 870], Faulty[670, 870]
 5: measure[4]:553, Phase[270, 770], Faulty[570, 670]
 6: measure[5]:589, Phase[170, 770], Faulty[570, 670]
 7: measure[6]:916, Phase[170, 770], Good
 8: measure[7]:925, Phase[170, 770], Good
 9: measure[8]:928, Phase[170, 770], Good
 10: measure[9]:941, Phase[170, 770], Faulty[570, 670]
 11: measure[10]:973, Phase[170,670], Good
 12: measure[11]:644, Phase[170,670], Good
 13: There is a fault in the cell: FF
 14: measure[12]:1673, Phase[170, 670], Good
 15: measure[13]:1683, Phase[170, 670], Good
 16: measure[14]:1693, Phase[170, 670], Faulty[570, 670]
 17: measure[15]:1700, Phase[170, 670], Faulty[570, 670]
 18: measure[16]:1707, Phase[170, 670], Good
 19: measure[17]:1701, Phase[170, 670], Faulty[570, 620]
 20: measure[18]:1711, Phase[170, 670], Faulty[570, 620]
 21: measure[19]:1719, Phase[170, 670], Faulty[570, 620]
 22: measure[20]:1718, Phase[170, 670], Good
 23: There may be a fault around the interconnection 1719
 24: interconnection(1719)
 25: MOS-FET(M8444)
 26: MOS-FET(M8412)

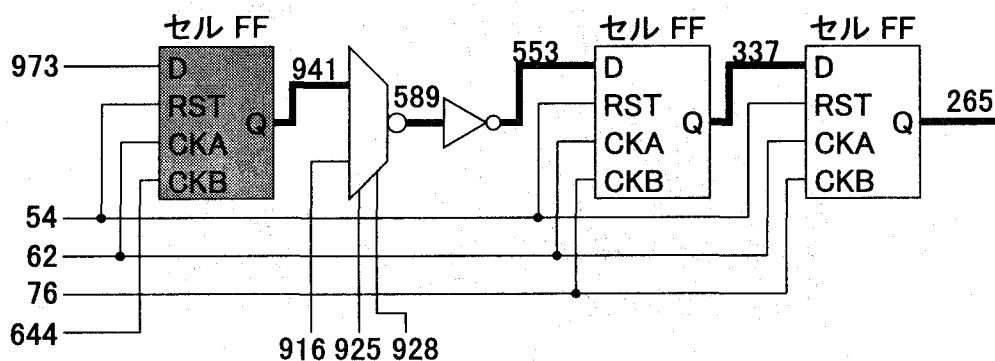


図 5.8 故障追跡によって、図 5.3 のレイアウトから抽出されたセルレベル回路図

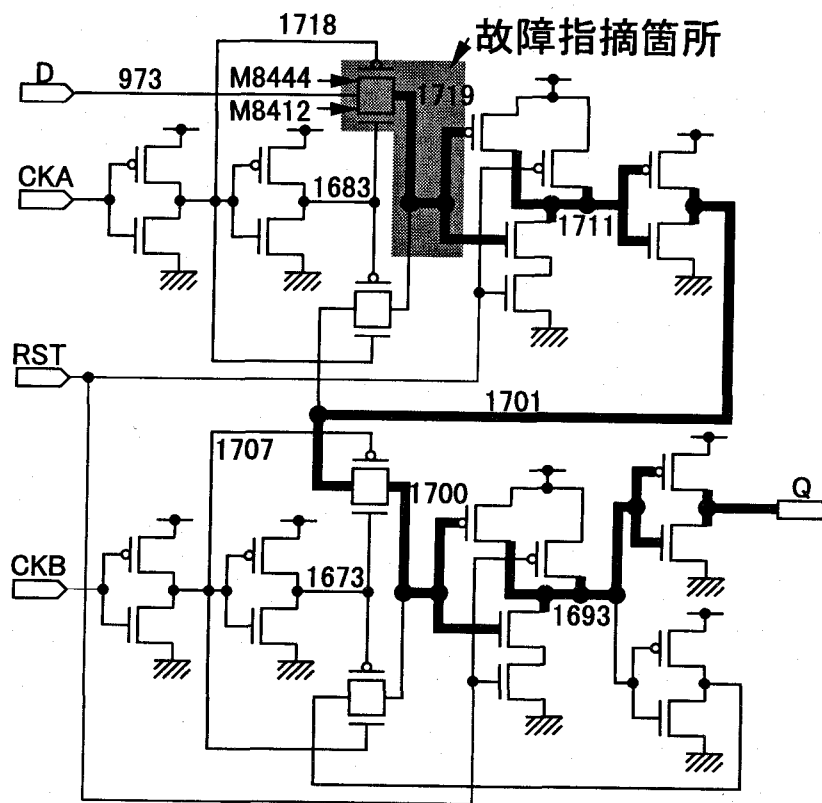


図 5.9 故障追跡によって、5.3 のレイアウトから抽出されたトランジスタレベル回路図

のような結果が出力される。図 5.8 と同じく、太線は故障追跡経路を示している。表 5.2 に示されているように、プロービング点数は 20 点である。また、総追跡時間は約 41 分である。このうち波形測定にかかった時間は全体の 49%、FAM 法による波形比較の時間は 14%、画像マッチングの時間は 11% である。これらの時間の比率は、プロービング点数には依存しない。従って、故障追跡の時間短縮の為に、プロービング点数の削減とともに、波形測定時間の短縮が重要である。

5.6 むすび

本章では、第 3 章、第 4 章の成果をふまえ、CAD レイアウトからの逐次回路抽出法および階層的故障追跡アルゴリズムを実装した、EB 自動故障追跡システムの構築について述べた。構築したシステムの特徴は以下の通りである。

- (1) EB テストシステム、LSI テストシステム、サーバ計算機、ホスト計算機をネットワークでリンクして、統合した構成である。
- (2) メイン、最適プロービング点選択、パターンマッチング、波形比較、EB テスタ制御、LSI テスタ制御の6つのプログラムを統合した故障追跡ソフトウェアを実装している。
- (3) 必要な CAD データはレイアウトデータのみである。
- (4) 始めに、SEM 画像と CAD レイアウト画像の位置合わせ、ノイズレベルを得る為の電源配線の波形測定、故障が発見された外部配線および故障信号の位相の入力はオペレータが行うが、以後の故障追跡は全て全自動で行える。
- (5) レイアウトデータの前処理時間は約2分であった(レイアウトデータサイズが130kB の場合。CAD レイアウトデータのサイズに依存する)。
- (6) 測定点1点あたりの波形測定および処理時間は約2分であった(マージン故障品で、DUT と良品の交換をする必要が無い場合)。

本システムを商用の CMOS 画像処理プロセッサ(トランジスタ数約54万個)の回路ブロック(トランジスタ数約4700個)に適用し、以下の結果を得た。

- (1) 故障追跡を自動的に行い、トランジスタレベルで故障箇所を正しく指摘することができた。
- (2) プロービング点数は20点であった。
- (3) 総追跡時間は約41分で、このうち、波形測定時間の占める割合が49%であった。

以上のことから、本システムが VLSI の故障追跡に有効であることが確認できた。また、本システムをさらに高速にする為には、プロービング点を削減する故障追跡アルゴリズムとともに、波形測定時間の短縮が重要であることが分かった。

第6章 結論

本研究により得た主たる成果を以下にまとめる。

(1) CAD レイアウトからの逐次回路抽出手法を提案した。本手法の特徴は、以下の通りである。

- (a) 相互にリンクされた回路データ構造体、四分木データ構造体等のデータ構造体を用いることで、高速に逐次的な回路抽出を行う。
- (b) 回路抽出処理を全体の前処理、逐次前処理、逐次抽出処理の3段階に分け、逐次前処理、逐次抽出処理は故障追跡アルゴリズムの要求に応じて実行する。故障追跡に関係の無い部分は抽出処理が行われないので、全回路抽出に比べ格段に処理量が少ない。

(2) VLSI の階層的故障追跡アルゴリズムを提案した。本アルゴリズムの特徴は、以下の通りである。

- (a) 信号の流れの向きに関する情報の含まれていないレイアウトデータからの抽出回路を解析し、信号の流れの向きを判定しながら故障追跡を行う。
- (b) 階層的に故障追跡を行い、トランジスタレベルで故障箇所の特定制を行う。
- (c) 電圧測定精度の低下する絶縁膜下の配線を回避して追跡可能である。
- (d) 絶縁膜への穴空け、パッド作成が可能な FIB 加工装置が利用可能であれば、EB テスタと併用して、より正確に効率良く故障箇所を絞り込むことができる。
- (e) テストシーケンスが長い順序回路においては、故障信号が測定される位相範囲のみを測定し、故障追跡に伴って測定位相を制御することによって、測定時間の短縮が可能である。
- (f) 順序回路を含んでいる場合も追跡可能である。

- (g) 双方向バス配線を含んでいる場合も追跡可能である。
- (3) 上述の CAD レイアウトからの逐次回路抽出手法および階層的故障追跡アルゴリズムを実装した、自動故障追跡システムを構築した。本システムの特徴は、以下の通りである。
- (a) EB テストシステム、LSI テストシステム、サーバ計算機、ホスト計算機をネットワークでリンクして、統合した構成である。
 - (b) メイン、最適プロービング点選択、パターンマッチング、波形比較、EB テスタ制御、LSI テスタ制御の 6 つのプログラムを統合した故障追跡ソフトウェアを実装している。
 - (c) 必要な CAD データはレイアウトデータのみである。
 - (d) 始めに、SEM 画像と CAD レイアウト画像の位置合わせ、ノイズレベルを得る為の電源配線の波形測定、故障が発見された外部配線および故障信号の位相の入力はオペレータが行うが、以後の故障追跡は全て全自動で行える。
 - (e) レイアウトデータの前処理時間は約 2 分であった(レイアウトデータサイズが 130kB の場合。CAD レイアウトデータのサイズに依存する)。
 - (f) 測定点 1 点あたりの波形測定および処理時間は約 2 分であった(マージン故障品で、DUT と良品の交換をする必要が無い場合)。
- (4) 自動故障追跡システムを商用の CMOS 画像処理プロセッサ (トランジスタ数約 54 万個) の回路ブロック (トランジスタ数約 4700 個) に適用した。得られた結果は以下の通りである。
- (a) 故障追跡を自動的に行い、トランジスタレベルで故障箇所を正しく指摘することができた。
 - (b) プロービング点数 20 点で、総追跡時間は約 41 分、このうち、波形測定時間の占める割合は約 49%であった。

- (c) 以上のことから、本システムが VLSI の故障追跡に有効であることが確認できた。また、本システムをさらに高速にする為には、プロービング点数を削減する故障追跡アルゴリズムと共に、波形測定時間の短縮が重要であることが分かった。

今後の課題として、以下の点が挙げられる。

- (1) 構築したシステムで故障追跡を行なう時、処理時間の多くは波形測定に費される。従って、システムを更に高速化するためには、故障追跡アルゴリズムの改良によってプロービング点数を削減すると共に、波形測定時間の短縮を図る必要がある。
- (2) 提案した階層的故障追跡アルゴリズムでは、信号の流れの向きを判定して上流に故障追跡するが、多数のトランSMIションゲートが接続されているバレルシフタなどの特殊な構成の回路では、信号の流れの向きを誤って判定する可能性がある。このような特殊な回路にも対処できるように、信号の流れの向きの判定アルゴリズムを改良し、判定精度の向上を図る必要がある。
- (3) 構築した自動故障追跡システムでは、100 万トランジスタ程度の階層構造レイアウトを取り扱うことができるが、近年の VLSI の中には、100 万トランジスタを越える大規模な設計のものがしばしば存在する。このような超大規模設計の VLSI の故障追跡を行なうため、システムの高速化、大容量化を図る必要がある。
- (4) 構築した自動故障追跡システムでは、DUT 配線パターンと CAD レイアウトのマッチングにより EB プロービング位置の補正を行なっているが、ボンディングワイヤによって生ずる磁界の影響で一次電子ビームが曲げられるなどして、プロービング位置がずれる可能性がある。システムの信頼性を向上するため、EB プロービング位置の補正法をさらに改良する必要がある。

謝辞

本研究の全過程を通じて、終始懇切な御指導、御教示ならびに御鞭撻を賜った大阪大学大学院工学研究科情報システム工学専攻 藤岡 弘教授に深甚なる感謝の意を表します。

また、本研究に関し、御指導、後教示を賜った大阪大学大学院工学研究科情報システム工学専攻 中前幸治助教授に心から謝意を表します。

本論文を執筆するにあたり、懇篤なる御指導を頂くと共に数々の御教示を賜った大阪大学大学院工学研究科情報システム工学専攻 白川 功教授、大阪大学大型計算機センター 村上孝三教授に深謝の意を表します。

さらに、論文執筆にあたり御指導と御高配を賜った大阪大学大学院工学研究科情報システム工学専攻 西尾章治郎教授、薦田憲久教授、鈴木 胖教授に厚く御礼申し上げます。

筆者が大阪大学大学院工学研究科電子工学専攻在学中に御指導を賜った、吉野勝美教授、裏 克己教授(現在、大阪産業大学教授)、濱口智尋教授、西原 浩教授、尾浦憲治郎教授、児玉慎三教授(現在、近畿大学教授)に深く感謝の意を表します。また、筆者が大阪大学大学院工学研究科電子工学専攻在学中ならびに大阪大学工学部情報システム工学科、同大学院工学研究科情報システム工学専攻在職中に御指導を賜った寺田浩詔教授(現在、高知工科大学教授)に深く感謝の意を表します。

第4章において、ISCAS'89 ベンチマーク回路のレイアウト生成に御協力頂いた大阪大学大学院工学研究科情報システム工学専攻 尾上孝雄助手ならびに佐藤 洋氏(現在、ローム株式会社)に深く感謝の意を表します。

第5章において、自動故障診断システムの構築に際し IDS Command Line Application Programming Interface を御提供頂いたシュルンベルジェ(株) 長友俊信氏、諸橋賢治氏に心より感謝の意を表します。

研究の遂行に際し協力ならびに助言を頂いた小迎 聡氏(現在、三洋電機株式会社)、杉本博司氏(現在、松下電器産業株式会社)、阪本 誉氏(現在、三菱重工業株

式会社), 田中裕久氏 (現在、松下電器産業株式会社)、研究に御協力頂いた中垣 亮氏 (現在、株式会社日立製作所)、松本 学氏 (現在、株式会社日立製作所)、西 孝啓氏 (現在、松下電器産業)、福留康和氏 (現在、関西電力株式会社) に深く感謝致します。

参考文献

- [1] W. Wolf, *Modern VLSI Design*, Prentice-Hall International, New Jersey, 1994.
- [2] 藤岡 弘, 中前幸治, “電子ビームテスター (1) 原理とハードウェア”, 応用物理, 技術ノート, **63**, 6, pp. 608–609 (1994).
- [3] 藤岡 弘, 中前幸治, 三浦克介, “特別招待論文: CAD リンク EB テストシステムによる VLSI 故障診断の新技法”, 電子情報通信学会技術研究報告 ICD95-187, pp. 21–28 (1995).
- [4] K. Ura, and H. Fujioka, “Electron Beam Testing,” *Advances in Electronics and Electron Physics*, ed. P. W. Hawkes, **73**, pp. 233–317 (1989).
- [5] E. Menzel, “Electron Beam Testing Techniques,” *Microelectronic Engineering*, **12**, 1–4, pp. 3–14 (1992).
- [6] N. Kuji, T. Tamama, and T. Yano, “A Fully-Automated Electron Beam Test System for VLSI Circuits,” *IEEE Design & Test*, October, pp. 74–82 (1985).
- [7] T. Tamama, and N. Kuji, “Automatic Fault Diagnostic EB Tester and Its Application to a 40K-Gate VLSI Circuit,” *Proc. 1985 International Test Conference*, pp. 643–649 (1985).
- [8] N. Kuji, T. Tamama, and M. Nagatani, “FINDER: A CAD System-Based Electron Beam Tester for Fault Diagnosis of VLSI Circuits,” *IEEE Transactions on Computer-Aided Design*, **CAD-5**, 2, pp. 313–319 (1986).
- [9] N. Kuji, and T. Tamama, “An Automated E-Beam Tester with CAD Interface,” *Proc. 1986 International Test Conference*, pp. 857–863 (1986).
- [10] S. Concina, G. Liu, L. Lattanzi, S. Reyfman, and N. Richardson, “Software Integration in a Workstation-Based E-Beam Tester,” *Proc. 1986 International Test*

- Conference, pp. 644–649 (1986).
- [11] F. Komatsu, M. Miyoshi, T. Sano, K. Sekiwa, and K. Okumura, “Electron Beam Tester Linked with a CAD Pattern Data,” Proc. XI International Congress on Electron Microscopy 1986, pp. 617–618 (1986).
- [12] S. Concina, G. Liu, and N. Richardson, “Designer-Oriented Interface to an E-Beam Tester,” 日本学術振興会第 132 委員会 第 97 回研究会 (EB テスティングシンポジウム/1986) 資料, pp. 46–51 (1986).
- [13] F. Komatsu, M. Miyoshi, T. Sano, and K. Okumura, “An Electron Beam Test System Linked with a CAD Database,” *Microelectronic Engineering*, **7**, 2–4, pp. 267–274 (1987).
- [14] S. Concina, and N. Richardson, “IDS 5000: an Integrated Diagnostic System for VLSI,” *Microelectronic Engineering*, **7**, 2–4, pp. 339–342 (1987).
- [15] S. Concina, and N. Richardson, “Workstation-Driven E-Beam Prober,” Proc. 1987 International Test Conference, pp. 554–560 (1987).
- [16] T. C. May, G. L. Scott, E. S. Meieran, P. Winer, and V. R. Rao, “Dynamic Fault Imaging of VLSI Random Logic Devices,” Proc. International Reliability Physics Symposium 1984, pp. 95–108 (1984).
- [17] J. P. Collin, D. Conard, B. Courtois, P. Denis, and D. Savart, “Failure Analysis Using E-Beam,” *Microelectronic Engineering*, **12**, 1–4, pp. 305–324 (1990).
- [18] J. D. Russell, F. L. Vargas, and B. Courtois, “E-Beam Testing Using Multiple Adjacent Image Processing for Prototype Validation,” *Microelectronic Engineering*, **16**, 1–4, pp. 413–420 (1992).
- [19] N. Kuji, and K. Matsumoto, “Marginal Fault Diagnosis Based on E-Beam Static Fault Imaging with CAD Interface,” Proc. 1990 International Test Conference, pp. 1049–54 (1990).

- [20] K. Nikawa, T. Nakamura, Y. Hanagama, T. Tsujide, K. Morohashi, and K. Kanai, "VLSI Fault Localization Using Electron Beam Voltage Contrast Image, — Novel Image Acquisition and Localization Method —," Jpn. J. Appl. Phys., **31**, Part 1, 12B, pp. 4525–4530 (1992).
- [21] 中村 豊一, 花釜 康子, 二川 清, 辻出 徹, 諸橋 賢治, 金井 健一, "高速電位像取得による故障箇所絞り込み手法の開発とその応用", 日本学術振興会第 132 委員会 第 121 回研究会 (EB テスティングシンポジウム/1992) 資料, pp. 104–109 (1992).
- [22] 小西 永二, 加藤 正次, 野口 和男, 平田 幸雄, 大金 秀治, 中村 豊一, 花釜 康子, 二川 清, 辻出 徹, "高速電位分布像の取得による故障箇所絞り込み手法を用いた LSI 解析事例の紹介", 日本学術振興会第 132 委員会 第 121 回研究会 (EB テスティングシンポジウム/1992) 資料, pp. 110–115 (1992).
- [23] A. Pindar, M. Ramanujam, C. Kardach, K. Morohashi, K. Kanai, and P. Cundall, "IFA: A Tool for Image-Based Failure Analysis on an E-Beam Diagnostic System," 日本学術振興会第 132 委員会 第 121 回研究会 (EB テスティングシンポジウム/1993) 資料, pp. 99–102 (1993).
- [24] 花釜 康子, 中村 豊一, 二川 清, 辻出 徹, 野口 和男, 加藤 正次, 平田 幸雄, 大金 秀治, "高速電位分布取得手法 (CGFI 法) におけるチャージアップの影響の低減", 日本学術振興会第 132 委員会 第 125 回研究会 (EB テスティングシンポジウム/1993) 資料, pp. 131–136 (1993).
- [25] 加藤 正次, 大金 秀治, 平田 幸雄, 花釜 康子, 中村 豊一, 二川 清, 小西 永二, 辻出 徹, "電子ビーム故障像法を用いた故障箇所絞り込み事例", 日本学術振興会第 132 委員会 第 125 回研究会 (EB テスティングシンポジウム/1993) 資料, pp. 137–142 (1993).
- [26] 浜田 弘幸, 辻出 徹, 菱井 利祐, "電子ビーム電位像を用いたメモリーの故障箇所特定化", 日本学術振興会第 132 委員会 第 125 回研究会 (EB テスティングシ

- ンポジウム/1993) 資料, pp. 143–148 (1993).
- [27] 中村 豊一, 加藤 正次, 小西 永二, 平田 幸雄, 大金 秀治, 諸橋 賢二, “動的故障像法を用いた LSI テスティング手法の開発”, 日本学術振興会第 132 委員会 第 128 回研究会 (LSI テスティングシンポジウム/1994) 資料, pp. 83–88 (1994).
- [28] 久慈 憲夫, 竹田 忠雄, “マルチチップモジュールの像モード故障診断”, 日本学術振興会第 132 委員会 第 128 回研究会 (LSI テスティングシンポジウム/1994) 資料, pp. 135–140 (1994).
- [29] S. Kochen, N. Landies, and D. Monson, “Computer-Guided Probing Techniques,” Proc. 1981 IEEE Test Conference, pp. 253–268 (1981).
- [30] 白川 千洋, 辺見 均, 松本 清, 寺本 光生, “EB テスタにおける故障追跡支援手法”, 日本学術振興会第 132 委員会 第 109 回研究会 (EB テスティングシンポジウム/1989) 資料, pp. 77–81 (1989).
- [31] 白川 千洋, 久慈 憲夫, “信号注入シミュレーションによる EBT ガイデッドプローブ”, 日本学術振興会第 132 委員会 第 117 回研究会 (EB テスティングシンポジウム/1991) 資料, pp. 59–64 (1991).
- [32] 白川 千洋, 久慈 憲夫, “EBT ガイデッドプローブによる 100k ゲート LSI の故障診断”, 日本学術振興会第 132 委員会 第 121 回研究会 (EB テスティングシンポジウム/1992) 資料, pp. 84–89 (1992).
- [33] A. C. Noble, “A Diagnostic Assistant for Integrated Circuit Diagnosis,” *Microelectronic Engineering*, **16**, 1–4, pp. 95–102 (1992).
- [34] N. Yamaguchi, T. Sakamoto, H. Nishioka, T. Majima, T. Satou, H. Shinada, H. Todokoro, and O. Yamada, “E-Beam Fault Diagnosis System for Logic VLSIs,” *Microelectronic Engineering*, **16**, 1–4, pp. 121–128 (1992).
- [35] A. C. Noble, “IDA: A Tool for Computer-Aided Failure Analysis,” Proc. 1992 International Test Conference, pp. 848–853 (1992).

- [36] 竹田 忠雄, 安部 哲哉, 下山 展弘, 中島 蕃, “EB テスタ (ガイドッドプローブ) を用いた大規模 ASIC の故障解析”, 日本学術振興会第 132 委員会 第 125 回研究会 (EB テスティングシンポジウム/1993) 資料, pp. 70–75 (1993).
- [37] S. W. Director, W. Maly, and A. J. Strojwas, *VLSI Design for Manufacturing: Yield Enhancement*, Kluwer Academic Publishers, Boston (1990).
- [38] A. Hu, and H. Nijjima, “New Approach to Integrate LSI Design Database with E-Beam Tester,” Proc. 1990 International Test Conference, pp. 1040–1048 (1990).
- [39] 則松 研二, 紫藤 真人, 石川 光昭, 藤井 美津男, “ゲートレベルナビゲーションシステムの開発と大規模回路への適用”, 日本学術振興会第 132 委員会 第 121 回研究会 (EB テスティングシンポジウム/1992) 資料, pp. 61–64 (1992).
- [40] 裏 克己, 藤岡 弘, 電子顕微鏡で観る LSI の世界, 日刊工業新聞社, 東京 (1990).
- [41] D. L. Crosthwait and F. W. Ivy, “Voltage Contrast Methods for Semiconductor Device Failure Analysis,” *Scanning Electron Microscopy* (1974).
- [42] K. Nakamae, H. Fujioka and K. Ura, “Local Field Effects on Voltage Contrast in the Scanning Electron Microscope,” *J. Phys. D: Appl. Phys.*, **14**, pp. 1939–1960 (1981).
- [43] K. Miura, K. Nakamae and H. Fujioka, “Automatic Transistor-Level Performance Fault Tracing by Successive Circuit Extraction from CAD Layout Data for VLSI in the CAD-Linked EB Test System,” *IEICE Trans. Electron.*, **E78-C**, 11, pp. 1607–1617 (1995).
- [44] K. Miura, K. Nakamae and H. Fujioka, “Hierarchical VLSI Fault Tracing by Successive Circuit Extraction from CAD Layout Data in the CAD-Linked EB Test System,” *Journal of Electronic Testing: Theory and Applications*, **10**, 3, pp. 255–269 (1997).

- [45] G. Kedem, "The Quad-CIF Tree: A Data Structure for Hierarchical On-Line Algorithms," Proc. of 19th Design Automation Conference, pp. 352–357 (1982).
- [46] S. M. Trimberger, *An Introduction to CAD for VLSI*, Kluwer Academic Publishers, Boston, 1987.
- [47] K. Miura, K. Nakamae and H. Fujioka, "Automatic Tracing of Transistor-Level Performance Faults with CAD-Linked Electron Beam Test System," IEICE Trans. Fundamentals, **E77-A**, 3, pp. 539–545 (March 1994).
- [48] K. Miura, K. Nakamae and H. Fujioka, "Hierarchical Fault Tracing for VLSIs with Bi-Directional Busses from CAD Layout Data in the CAD-Linked EB Test System," IEICE Trans. Electron., **E80-C**, 3, pp. 498–502 (March 1997).
- [49] K. Miura, K. Nakamae and H. Fujioka, "Hierarchical Fault Tracing for VLSI Sequential Circuits from CAD Layout Data in the CAD-Linked EB Test System," Proc. Asia and South Pacific Design Automation Conference 1997, pp. 329–332 (1997).
- [50] Y. Mashiko, H. Morimoto, H. Koyama, S. Kawazu, T. Kaito and T. Adachi, "A New VLSI Diagnosis Technique: Focused Ion Beam Assisted Multi-Level Circuit Probing," Proc. International Reliability Physics Symposium, pp. 111–117 (1987).
- [51] K. Nikawa, H. Hosoi, M. Murase, N. Hirayama, K. Nasu, H. Yuasa and S. Inoue, "Activities to Improve Failure Analysis Ability in an LSI Manufacturer," Proc. International Conference on Reliability and Maintainability, pp. 510–515 (1988).
- [52] C. G. Talbot, S. Concina, D. Masnaghetti, K. Rao, and N. Richardson, "Linked E-Beam Probing and Focused Ion Beam Technology," 日本学術振興会第 132 委員会 第 117 回研究会 (EB テスティングシンポジウム/1991) 資料, pp. 109–113 (1991).

- [53] 岩崎 浩二, 大井 将道, 佐藤 誠, 一宮 豊, 皆藤 孝, 足立 達也, “FIB-EB 複合装置とその応用”, 日本学術振興会第 132 委員会 第 125 回研究会 (EB テスティングシンポジウム/1993) 資料, pp. 158-163 (1993).
- [54] 坂田 隆英, 熊谷 清人, 坂口 清志, 後藤 俊徳, 小久保 靖, “FIB/SEM 複合装置とその応用例”, 日本学術振興会第 132 委員会 第 128 回研究会 (LSI テスティングシンポジウム/1994) 資料, pp. 159-163 (1994).
- [55] D. V. Heinbuch, *CMOS3 Cell Library*, Addison-Wesley Publishing Company, New York, 1988.
- [56] K. Miura, K. Nakata, K. Nakamae and H. Fujioka, “Automatic EB Fault Tracing System by Successive Circuit Extraction from VLSI CAD Layout Data,” Proc. 6th Asian Test Symposium, to be published.
- [57] 中垣 亮, 松本 学, 中前幸治, 藤岡 弘, “EB テストシステムにおけるプロービング点自動決定アルゴリズム”, 日本学術振興会第 132 委員会 第 128 回研究会 (LSI テスティングシンポジウム/1994) 資料, pp. 123-128 (1994).
- [58] 福留康和, 中前幸治, 藤岡 弘, “ハイブリッド GA による CAD レイアウトと DUT 配線パターン観測画像とのパターンマッチング”, LSI テスティングシンポジウム/1996 会議録, pp. 24-29 (1996).
- [59] 西 孝啓, 中前幸治, 藤岡 弘, “EB テストシステムにおける信号波形比較システム”, 日本学術振興会第 132 委員会 第 132 回研究会 (LSI テスティングシンポジウム/1995) 資料, pp. 126-131 (1995).