OUKA
Osaka University Knowledge Archive

Title	全可観測な環境での論理回路のテスト容易化設計に関する研究
Author(s)	温, 暁青
Citation	大阪大学, 1993, 博士論文
Version Type	VoR
URL	https://doi.org/10.11501/3065910
rights	
Note	

Osaka University Knowledge Archive : OUKA

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

Osaka University

全可観測な環境での
論理回路のテスト容易化設計に関する研究

1993年1月

温　暁　青

全可観測な環境での
論理回路のテスト容易化設計に関する研究

1993年1月

温　暁　青

内容梗概

正しく設計された集積回路であっても，製造段階で何らかの欠陥が生じる可能性がある．また開発段階では，正しい設計を得るために試作回路について故障原因を究明することが必要である．集積回路のテスト，つまり，製造された回路が設計通りに動作するか否かを調べる故障検査，および設計通りに動作しない回路の不良箇所を指摘する故障診断は，集積回路の開発にとって重要な役割を果たしている．

近年，集積回路の大規模化と高機能化につれて，故障検査はますます困難になってきている．まず，故障検査に必要なテストベクトルの生成においては，回路規模の増大により生成時間が大きな問題となる．テスト生成ができたとしても，通常その数が膨大であるため，多量生産される集積回路のテストにとって，テストの実施時間も大きな問題となる．また，CMOS 回路技術が主流になりつつあるため，CMOS 回路に特有の新しい故障に対する検査方法の開発も重要な課題となる．

故障診断は一般的に故障検出よりはるかに困難である．最近では，特定用途向け集積回路（Application Specific IC: ASIC）の多用により，故障診断が以前より頻繁に行なわれるので，迅速かつ効率的な故障診断手法の開発も急務となっている．

集積回路のテストの問題を根本的に解決するために，従来からテスト容易化設計というアプローチが用いられている．その基本的な考え方は，回路設計後にテストを考えるのではなく，テスト容易な回路を設計することである．その代表的な例は順序回路のスキャン設計であり，順序回路の故障検査をそれより簡単な組合せ回路の故障検査におきかえるものである．

故障診断では，従来から回路の内部状態を観測するプロービング技術が用いられて来ている．最近では，電子ビームプロービング技術と CAD システムの統合による高度な故障診断システムである電子ビームテスタが開発されている．電子ビームプロービング技術を用いることにより回路内の大部分の信号線が観測できるため，故障診断は容易になる．しかし，電子ビームプロービング技術は，短いテスト系列を用いるときにのみ有効である．任意の論理回路が短いテスト系列をもつ保証がないため，テスト容易化設計を施して回路が短いテスト系列をもつようにすることが，電子ビームプロービング技術を実現するために必要である．

電子ビームプロービング技術に基づいて，回路内のすべてのゲートの出力線が観測できると仮定するテスト環境を全可観測な環境と呼ぶ．本論文は，全可観測な環境におけるテスト容易な論理回路の特徴，性質および構成法についての研究成果をまとめたものである．

第 1 章では，本研究の背景や目的について述べている．

第 2 章では，研究の対象回路とする論理回路の基本概念，および論理回路のテストとテ

スト容易化設計について紹介している．

第 3 章では，全可観測な環境を実現する電子ビームプロービング技術と，全可観測な環境におけるテスト容易化設計の必要性について述べている．

第 4 章では，全可観測な環境でのテスト容易な組合せ回路である k - UCP 回路の概念と故障検査について述べている．k - UCP 回路は NOT ゲートと k 入力の基本論理ゲートで構成される．k - UCP 回路において，長さ $k+1$ のテスト系列ですべての縮退故障を検出することができ，長さ $k(k+1)+1$ のテスト系列ですべてのスタック・オープン故障を検出することができる．さらに，k - UCP 回路の拡張として，k - R 回路を提案している．

第 5 章では，組合せ回路を k - UCP 回路に変換する手法について述べている．この変換での付加ゲート数を最小にする方法を見つけることは NP 完全問題なので，ヒュリスティックな手法を用いてその解決を試みた．実験結果は，k 入力 NAND または NOR ゲートからなる回路を k - UCP 回路に変換するためのオーバーヘッドが少ないことを示している．

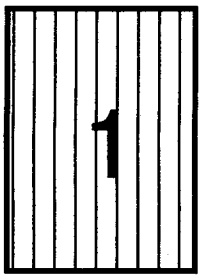
第 6 章では，全可観測な環境での順序回路のテスト容易化設計として，k - UCP 順序回路および k - UCP スキャン回路を提案している．いずれの順序回路の組合せ部分も k - UCP 回路である．しかし，普通の順序回路の場合，k - UCP 回路に変換された組合せ部分に所要のテストベクトルを繰り返し印加することは困難かまたは不可能である．k - UCP 順序回路においては回路内のフリップ・フロップの入力側にゲートを挿入することにより，また k - UCP スキャン回路においてはスキャンパスの組み方を工夫することにより，組合せ部分へのテストベクトルの繰り返し印加を容易にしている．それぞれの回路においては，長さ $3(k+1)$ と $k+1$ のテスト系列ですべての縮退故障をテストすることができることを示している．

第 7 章では，テスト容易な順序回路への変換方法について述べている．実験結果は，スキャンパスをもち，かつ組合せ部分が k 入力 NAND または NOR ゲートのみを含む順序回路を k - UCP スキャン回路に変換するためのオーバーヘッドが少ないことを示している．

第 8 章ではまず，CMOS の k - UCP 回路のスタック・オープン故障の診断手法について述べている．k - UCP 回路にテスト集合を印加したとき，故障ゲートの出力線および他のいくつかの信号線に異常系列が現われる．これらの異常系列の位置と異常系列が基本系列と異なるビットの位置を用いて故障診断を行なう手法を示している．また，一般の論理回路の故障診断にも適用できる電子ビームテスタに基づくガイディド・プローブ法による 2 段階の故障診断過程を提案している．第 1 段階では，最上位層にある信号線のみを観測点の候補とし，高速な観測点決定手法を用いて被疑部分回路を効率よく絞り込む．第 2 段階では，下位層にある信号線をコストの高い FIB で露出させる必要があるので，最小の平均観測回数が得られる観測点決定手法を用いる．その結果，回路全体の故障診断を効率よく行なうことができる．

第 9 章では，本研究のまとめと今後の課題について述べている．

目　次



序論

1.1 研究の背景

集積回路（Integrated Circuit: IC）は，シリコン基板内にダイオード，トランジスタ，抵抗および容量を作り込み，これらを総合配線して電子回路を構成したものである．その特徴は，小型，高速，省電力および高信頼性である．高性能マイクロプロセッサや大容量メモリなどに代表されるように，集積回路は近代の産業と社会に計り知れないほど大きな影響を与えており，将来にもなくてはならない存在として発展していくものと思われる [1].

集積回路は60年代の初期に発明されて以来，集積度と機能面において急速な発展を遂げてきた．初期は集積回路内のトランジスタがわずか2個であったが，現在は回路の最小線幅がコンマ数ミクロンになっており，数百万ものトランジスタを搭載する集積回路が製造されている．機能面においては，簡単な論理演算しかできない集積回路から複雑なシステムを搭載する集積回路へと進歩してきた．

集積回路の急速な進歩を支えてきているのは設計技術，製造技術およびテスト技術である．設計技術と製造技術は，与えられた回路仕様に対して自動的に設計を行なう自動合成システムやコンマ数ミクロン線幅の回路パターンを正確に扱うサブミクロン級微細加工技術に代表されるように，この数十年で大きな進歩を遂げてきた．ところが，設計技術と製造技術の目覚ましい進歩に比べて，テスト技術は必ずしも大幅な進展をしてこなかった．その結果，テストは集積回路の発展にとって主要な問題点になっている [2-4].

集積回路のテストとはその故障検査と故障診断の両方を意味する．故障検査とは，回路が設計通りに動作するか否かを調べることである．故障診断とは，設計通りに動作しない回路に対し，その原因を突き止めることであり，その中心は不良箇所を指摘することである．

故障検査

故障検査ではまず，すべての対象故障に対して，正常時と故障時の回路の出力値が異なるような入力ベクトルを生成する．これをテスト生成といっている．生成したテストベクトルを回路に印加し回路の応答値を観測する．応答値が期待値と異なれば，回路に故障があるものと判断される．すべてのテストベクトルに対する応答値が期待値と一致すれば，回路が正

常であると認められる [5-7]. これをテストの実施といっている.

テスト生成では, 対象故障に対して故障箇所でその影響を顕在化させ, 観測のためにさらにその故障の影響を外部出力線まで伝搬させる必要がある. 大規模または複雑な回路の場合, テスト生成時間は長すぎる場合がある. テスト生成ができても, テストベクトル数が膨大になった場合, それらを保存するために多くのメモリ容量が必要であるし, テストの実施時間も長くなり, 多量生産にとって深刻な問題となる [2-4].

故障検査における諸問題を解決するアプローチとして, テスト生成アルゴリズムの効率化およびテスト容易化設計などが知られている.

Dアルゴリズム [8] が提案されて以来, 組合せ回路に対して多くのテスト生成アルゴリズムが提案されている. 現在, PODEM [9], FAN [10], CONT [11], SOCRATES [12] などが知られている. しかし, 順序回路に対しては, 一般的に有効なものは提案されていない [13].

テスト生成アルゴリズムを改良するだけでは, 故障検査における諸問題を根本的に解決することができない. そこで, 回路設計後にテストを考えるのではなく, 設計段階においてもテストへの考慮を取り入れてテスト容易な回路を設計する, いわゆるテスト容易化設計 (Design For Testability) が研究されている [14-20]. その代表的な例は順序回路の故障検査を組合せ回路の故障検査におきかえるスキャン設計である. テスト容易化設計の導入により, 回路仕様の実現にとって余分なハードウェアや動作速度の低下などのオーバーヘッドが生じることがある. 以前はハードウェアがテスト時間より高価であったため, なるべく少ないハードウェアで設計を行なう傾向があった. しかし最近では, 集積技術の発達により, ハードウェアのコストが著しく安くなってきており, 回路仕様の実現にとって余分なハードウェアの付加を伴うテスト容易化設計の導入が現実的なアプローチとなっている. また, 大規模な回路の故障検査に必要な時間が増加する一方で, ハードウェアを導入してもテストを容易にすることが必要になってきている.

故障診断

故障診断は, 集積回路の開発にとって重要な意味をもつ. その主な理由として三つが挙げられる. 第1は, 開発段階での故障原因の究明である. 集積回路の開発では, 機能の異常, 特性の不良, マージン不良など, 数多くの問題が生じる. 量産へ移行できる設計・プロセスを確立するためには, 故障原因を究明するための診断・設計の修正・試作というサイクルを繰り返すことが余儀なくされている. 特に, 集積回路の主流となりつつある特定用途向け集積回路の場合, 開発期間の短縮と開発コストの低減が必須なので, 設計・開発を効率化するため, 迅速かつ的確な故障診断が不可欠となる. 第2は, 量産段階において発生する不良の原因究明である. 歩留りなどが低下した場合, 原因を明らかにし, 設計やプロセス上の対策を立てる必要がある. 第3は, 市場で発生した故障の原因究明である. システムが大規模化し, 1個の集積回路の故障が重大な影響を及ぼすため, 早急な対策が必要である.

故障診断は故障検査と異なって，回路が正常か異常かのチェックのみではなく，異常の原因を突き止めることを目的とするため，故障検査よりも困難であることは明らかである．特に，外部出力線しか観測できない場合，故障診断ができるのは極めて小さい回路に限られてしまう．そこで，従来から回路の内部状態をプロービングにより観測する技術は故障診断にとって不可欠なものとなっている．

プロービング技術は接触系と非接触系に分けられる．最初に使用されていたのは金属の針によるプロービングであった．この方法では，回路の電気的な特性および回路構造そのものを破壊する恐れがあるので，集積度の高い集積回路には適用できない．非接触系のプロービング技術では電子ビームなどによるプローブを利用するため，回路の電気的な特性や構造を壊す恐れがなく，集積度の高い集積回路にも適用できる [23-26]．電子ビームプロービング技術を利用するため，集積回路の封装を壊す必要があるので，その使用は故障診断に限られている．

以上，集積回路の故障検査と故障診断の問題点とこれまでの解決の試みについて述べた．基本的には，これまでの解決法を三つのグループに大別することができる．第 1 は外部出入力線しか観測・制御できないという通常のテスト環境で効率化を図る方法である．第2は回路構造を変更するテスト容易化設計である．第 3 はテスト環境を変える方法であり，その代表は電子ビームプロービング技術に基づくテスト方法である．

1.2 研究の目的

本論文では，外部出力線しか観測できないようなテスト環境を通常のテスト環境と呼び，すべてのゲートの出力線が観測できるようなテスト環境を全可観測なテスト環境，あるいは単に全可観測な環境と呼ぶ．

これまでは，テスト容易化設計と全可観測な環境でのテストに関する研究が独立的に進められてきた．つまり，テスト容易化設計は通常のテスト環境で行なわれ，全可観測な環境ではテスト容易化設計が施されていない．これは，通常のテスト環境でのテスト生成に比べ，全可観測な環境でのテスト生成は簡単であるからである．しかし以下の理由で，全可観測な環境においてもテスト容易化設計を行なう必要がある．

全可観測な環境の基礎となる電子ビームプロービング技術では，高速動作中の回路の内部状態を観測するために，ストロボ法が用いられる [27]．そのため，テスト系列を回路に繰り返し印加することが必要である．また，高速かつ正確な観測を行なうために，回路を短い周期の周期動作状態にする必要がある [28,29]．つまり，電子ビームプロービング技術で用いられるテスト系列は二つの条件を満たさなければならない．第 1 はテスト系列が短いことであり，第 2 はそのテスト系列を回路に繰り返し印加することができることである．全可観測な環境で生成されたテスト系列は，電子ビームプロービング技術を用いたテスト環境で

使用できるほど短くない場合もある．また，順序回路においては，テスト系列が短くても，それを回路に繰り返し印加することができない場合が多い．したがって，全可観測な環境においても，回路が短くてかつ繰り返し印加することができるテスト系列をもつように回路設計を工夫する必要がある．つまり，全可観測な環境においてもテスト容易化設計が必要である．

さらに，電子ビームプロービング技術では，1本の内部信号線の状態を測定するために相当長い時間がかかる．多くの信号線を観測しなければならない場合に，故障診断の所要時間が長くなる．すなわち，なるべく少ない観測回数で故障診断を行なう必要がある．これも，全可観測な環境においてもテスト容易化設計が必要な理由である．

このように，全可観測な環境においてもテスト容易化設計が必要である．しかしこれまでは，この方面の研究は少ない．短いテスト系列でテストできるように論理回路を変換する方法として，組合せ回路を適当に変換することにより，5個 [34] または3個 [35] のテストベクトルでテストできることが示されている．しかし，これらの方法では，信号線の縮退故障のみを対象故障としている他，回路を構成する素子の種類や入力数についての制限が厳しい．また，回路変換は主に局所的な置換で行なわれ，オーバーヘッドが大きいと思われる．そこで，より広範的，系統的なアプローチを取るべく，本研究では以下のことを具体的な目標とする．

(1) 全可観測な環境において，短いテスト系列を有する組合せ回路と順序回路の概念を提案する．信号線の縮退故障の他に，スタック・オープン故障をも対象故障とする．また，生成したテスト系列を回路に繰り返し印加することができるようにする．

(2) 提案したテスト容易な論理回路の故障検査と故障診断の手法を示す．

(3) 任意の論理回路から提案したテスト容易な論理回路への効率的な変換手法を提案する．また，ベンチマーク回路を用いて実験を行ない，その有効性を確かめる．

(4) 全可観測な環境において，一般の論理回路の故障診断に対しても効率的な手法を提案する．

1.3 論文の構成

本論文は次のように構成されている．第1章では，本研究の背景，目的および構成について述べる．第2章では，論理回路のテストとテスト容易化設計の基本概念について簡単に述べる．第3章では，回路の内部状態を観測する電子ビームプロービング技術について述べ，それに基づく全可観測な環境においてもテスト容易化設計が必要である理由を明らかにする．第4章では，全可観測な環境での組合せ回路のテスト容易設計である k-UCP 回路と k-R 回路の概念およびテスト手法について述べる．第5章では，任意の組合せ回路を k-UCP 回路に変換する手法について述べる．第6章では，全可観測な環境での順序回路の

テスト容易設計である k-UCP 順序回路および k-UCP スキャン回路の概念およびテスト手法について述べる．第 7 章では，任意の順序回路をテスト容易な順序回路に変換する手法について述べる．第 8 章では，k-UCP 回路のスタック・オープン故障の効率的な故障診断手法，および一般の論理回路の故障診断にも使用できるガイディド・プローブ法の効率化について述べる．第 9 章では，本論文のまとめと今後の課題について述べる．

2 論理回路のテストとテスト容易化設計

本章では，本論文で対象回路とする論理回路，論理回路のテストおよび論理回路のテスト容易化設計に関する基本概念について説明する．

2.1　論理回路の概念

集積回路は使用するトランジスタの種類により，バイポーラ回路とMOS回路に分類される．前者は一般に高速であるが消費電力が大きく，製造工程が複雑でかつ大きなチップ面積を必要とするのに対して，後者はほぼその反対であり，大規模な集積回路にMOSが多用されている．特に，MOS集積回路の中でCMOS集積回路が低消費電力用として一般に用いられている [1].

集積回路はその内部信号の取り扱い方により，アナログ集積回路とディジタル集積回路に大別される．前者はA/D，D/Aの相互変換を含み，後者は論理演算を行なう論理回路と論理値を記憶するメモリに分かれる．アナログ集積回路よりディジタル集積回路の方が圧倒的に多いが，これは集積回路のデジタル適性に基づく [1].

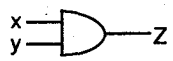
論理回路とは，入力値，出力値および内部状態の値が論理値0または1の組合せとして表現できる回路である．論理回路はさらに組合せ回路と順序回路に分類される [5-7, 51]．組合せ回路は，出力値がそのときの入力値により一意的に決まる回路である．すなわち，その出力は入力変数のブール関数で表すことができる．一方，順序回路は，現在の入力値のみならず，過去にどのような入力値がどのような順序で加えられたかも現在の出力値に影響を与える．

一般に，論理回路を記述する方法としては，自然言語による記述，VHDLなどのハードウェア記述言語による記述，真理値表による記述，論理式による記述，基本論理素子による記述，トランジスタ，抵抗，容量などによる記述，レイアウト情報による記述などがある．以下では，基本論理素子による記述とトランジスタによる記述について説明する．

2.1.1　基本論理素子による記述

基本論理素子は論理ゲートと記憶素子を含む．論理ゲートとは，AND, OR, NAND, NOR, NOT, XOR, XNOR などの基本的な論理演算を行なう回路のことである．主な論理ゲートの機能と記号を表 2.1 に示す．

表 2.1　主な論理ゲートの機能と記号

ゲート名	記号	論理式
AND		$Z = x \wedge y$
OR		$Z = x \vee y$
NAND		$Z = \overline{x \wedge y}$
NOR		$Z = \overline{x \vee y}$
NOT		$Z = \overline{x}$
XOR		$Z = x \wedge \overline{y} \vee \overline{x} \wedge y$
XNOR		$Z = x \wedge y \vee \overline{x} \wedge \overline{y}$

組合せ回路は AND, OR, NAND, NOR, NOT, XOR, XNOR などの論理ゲートを用いて実現できる．実際には，NAND ゲートまたは NOR ゲートを用いるだけでも，任意の組合せ回路を実現することができる [51]．組合せ回路の例を図 2.1 に示す．この回路の入力値は 0, 1, 0 である．回路の出力値はその入力値により一意的に決まって 1 である．

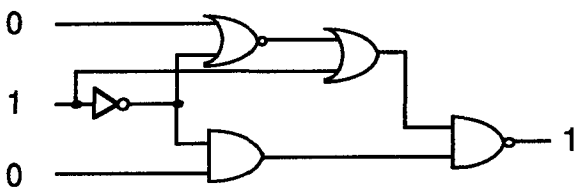


図 2.1　組合せ回路の例

順序回路を表現するため，論理ゲートの他に，内部状態を記憶する素子も必要である．図 2.2 はもっとも基本的な記憶素子である *D*- フリップ・フロップの記号を示す．

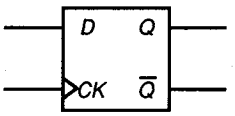


図 2.2　*D*- フリップ・フロップの記号

図 2.2 では，*CK* より同期用のクロック・パルスが印加される．クロック・パルスがないとき，*Q* の値が保持される．クロック・パルスが印加されると，*Q* が *D* の値となる．回路全体の動作がクロック・パルスで同期される順序回路が同期式順序回路と呼ばれている．図 2.3 に同期式順序回路の例を示す．この回路の現在の出力値は，現在の外部入力値の 0 と 1 だけではなく，フリップ・フロップの出力値である内部状態にも依存する．

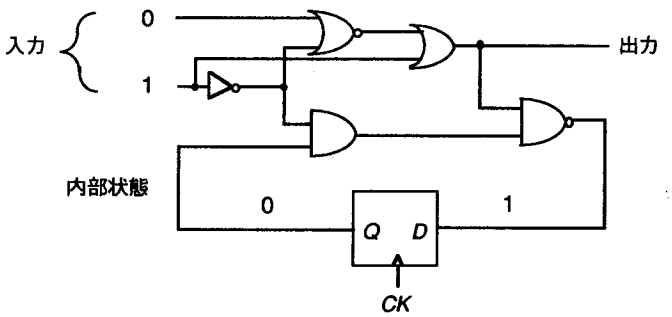


図 2.3　同期式順序回路の例

2.1.2　トランジスタによる記述

ここでは，CMOS 回路の場合について説明する．CMOS 回路では，2 種類の MOS トランジスタ，*n* トランジスタと *p* トランジスタ，が使用される [1]．*n* トランジスタと *p* トランジスタをスイッチと見なすことができる．これを図 2.4 に示す．

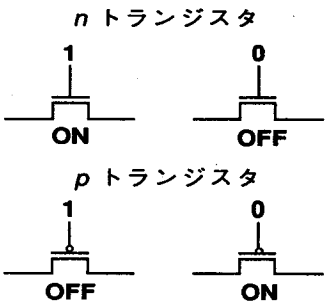


図 2.4　MOS トランジスタ

図 2.5 から図 2.7 に CMOS の 2 入力 AND, OR, NAND, NOR および NOT ゲートのトランジ

スタによる記述を示す．図2.8にCMOSゲートの一般構成を示す．CMOSゲートの特徴は，正常時には，pトランジスタ側とnトランジスタ側の両方とも導通状態または非導通状態にならないことである．

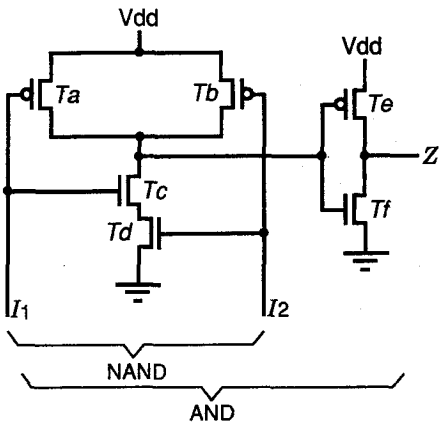


図2.5　CMOSの2入力 NANDと ANDゲート

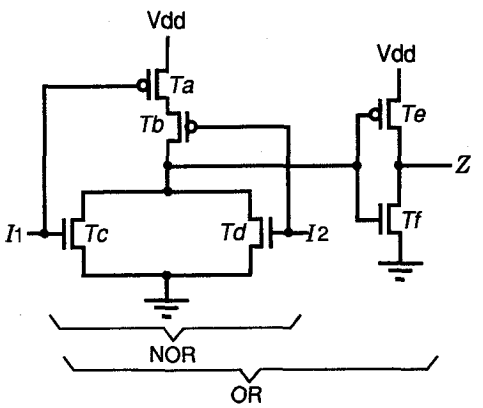


図2.6　CMOSの2入力NORとORゲート

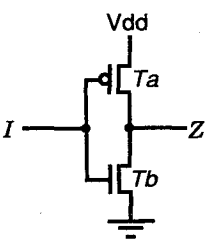


図2.7　CMOSのNOTゲート

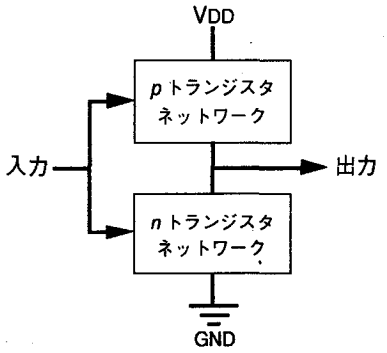


図 2.8　CMOS ゲートの一般構造

本論文では，組合せ回路と D-フリップ・フロップを記憶素子とする同期式順序回路を対象回路とする．また，低消費電力で大規模集積回路に適したCMOS技術が論理回路の主な製造技術となってきているため，本論文ではトランジスタによる記述の論理回路とはCMOS論理回路を指すものとする．

2.2　論理回路の設計と製造

通常，論理回路の設計は複数の段階に分けて行なわれる．図2.9に主な設計段階を示す．

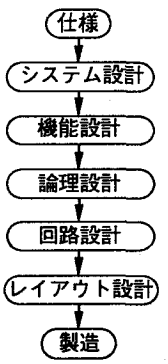


図 2.9　論理回路の設計の流れ

このような階層化設計の流れは回路データの変換の流れと見なすことができる．すなわち，ある段階での設計というのは，与えられた上位の設計データを下位のより詳細な設計データに変換することである．この設計段階の最上位の設計データはユーザからの要求であり，最下位の設計データは直接に製造に使用されるレイアウト・データである．

各設計段階では，回路を表現する方法が異なる．下位に下がるにつれて，回路表現が詳細になってくる．仕様設計では，自然言語に近いもので回路記述を行なう．機能設計では，ALU，レジスタやメモリを用いて回路を表現する．論理設計では，論理ゲートやフリップ・

フロップなどで回路を表現する．回路設計では，トランジスタ，抵抗や容量などを用いて回路記述を行なう．最後にレイアウト設計では，金属，ポリシリコン，コンタクトなどを表す長方形で回路を表現する．このレイアウト・データを用いて製造が行なわれる．

2.3　論理回路のテスト

正しく設計された回路であっても，製造段階で様々な故障が導入され，出来上がった回路が設計通りに動作しない場合がある．故障の原因は，シリコン基板の汚れ，マスクのずれ，アルミ配線の腐食などと様々である．その結果として，回路に断線やショットなどの物理故障が起こり，その影響が外部出力線での誤動作として現われる．集積回路の量産段階や納品検査のときなどは，正常に動作しない回路は破棄されるが，開発段階において正しい設計を得るためまたは量産段階において歩留まりを高めるために，正常に動作しない回路に対しその原因を突き止めることが必要である．つまり，回路が設計通りに動作する否かを調べる故障検出および設計通りに動作しない回路の故障原因を突き止める故障診断が必要である．論理回路のテストとはその故障検出および故障診断の両方を意味する．

故障検査ではまず，すべての対象故障に対して，正常時と故障時の回路の出力が異なるような入力ベクトルを生成する．これをテスト生成といっている．生成したテストベクトルを回路に印加し回路の応答値を観測する．応答値が期待値と異なれば，回路に故障があるものと判断される．すべてのテストベクトルに対する応答値が期待値と一致すれば，回路が正常であると認められる [5-7]．故障診断においては，テスト生成の他に，故障位置を特定する手法も必要となる．

以下では，テスト生成における故障モデルの概念，故障検査用テスト生成アルゴリズム，および故障診断の主な手法について説明する．

2.3.1　故障モデル

テスト生成では，故障をどのように取り扱うかが重要である．まず，故障を意識しない生成法がある．例えば，n 入力の組合せ回路は 2^n 個の可能な入力ベクトルをもち，それらをすべて印加することにより，その回路の論理機能が設計通りであるか否かを調べることができる．しかし，n が少しでも大きくなると，2^n は膨大な数になり，実際には使えなくなる．したがって，より現実的なテスト生成方法では，回路に対して対象故障を仮定してから，すべての対象故障が回路に存在するかどうかを調べることにより故障検出を行ない，どの対象故障が存在するかを調べることにより故障診断を行なう．

対象故障の決め方として，二つの方法が考えられる．第 1 の方法では，物理的に起こりうるすべての故障を対象故障とする．しかし，現実的には，物理的に起こりうるすべての故障をあらかじめ調べることが不可能であったり，対象故障の数が莫大であったりするために，物理的に起こりうる故障を直接に取り扱うことが不可能に近い．第 2 の方法では，多

種多様な物理的故障の抽象化である故障モデルが用いられている．

2.3.1.1　故障モデルの概念

故障モデルは故障の影響，つまり故障が信号値にもたらす変化を表すものである．故障モデルの使用により以下の利点が得られる [5-7].

(1) 起こりやすい物理故障に対するテストベクトルを生成することができる．

(2) テスト生成の効率化をはかることができる．

(3) 故障診断が簡単になる．

各設計レベルでは異なる故障モデルを考えるのが普通である．各設計レベルとそれに対応する故障モデルを表 2.2 に示す．

表 2.2　設計レベルと故障モデル

回路記述	故障モデル
レジスタ転送レベル	機能故障
ゲートレベル	縮退故障 遅延故障 ブリッジ故障
トランジスタレベル	スタック・オン故障 スタック・オープン故障
レイアウトレベル	断線　　短絡 クロストーク

現在よく使用されている故障モデルは，ゲートレベルの縮退故障モデルとブリッジ故障モデル，およびトランジスタレベルのスタック・オープン故障モデルとスタック・オン故障モデルである．本論文では，故障モデルで仮定される故障を単に故障と呼ぶことにする．以下では，本論文で対象故障とする縮退故障とスタック・オープン故障について説明する．

2.3.1.2　縮退故障

縮退故障は従来から用いられてきたゲートレベル回路における故障モデルである [5-7]. これは信号線が論理値 1 または 0 に固定される故障を表す故障モデルである．論理 1 に固定される縮退故障を 1 縮退故障と呼び，論理値 0 に固定される縮退故障を 0 縮退故障と呼ぶ．例えば，信号線が電源にショートされた故障は 1 縮退故障でモデル化できる．

図 2.10 に示す 2 入力 NAND ゲートにおいては，6 個の縮退故障が考えられる．つまり，入力線 a と b の 1 縮退故障と 0 縮退故障，および出力線 Z の 1 縮退故障と 0 縮退故障である．しかし，a と b の 0 縮退故障は任意の入力ベクトルに対して Z の 1 縮退故障と同じ値

を出力するので，この3個の故障は等価故障である．その中の1個の故障，例えば，Zの1縮退故障を対象故障とすればよい．このときのZの1縮退故障が代表故障と呼ばれている．したがって，2入力NANDゲートの場合，入力線aとbの1縮退故障，および出力線Zの1縮退故障と0縮退故障を対象故障とすれば十分である．

図2.10　2入力NANDゲート

1個の縮退故障を検出するために，1個のテストベクトルが必要である．例えば，2入力NANDゲートの入力線aの1縮退故障故障を検出するには，$<a=0, b=1>$を印加すればよい．これは，正常時のゲートの出力は1であるが，故障時のゲートの出力は0であるからである．また，1個のテストベクトルで複数の縮退故障を検出することも可能である．例えば，$<a=0, b=1>$で，aの1縮退故障故障と出力線Zの0縮退故障を検出することができる．2入力NANDゲートのすべての縮退故障を検出するために，$<a=1, b=0>$, $<a=0, b=1>$および$<a=1, b=1>$という3個のテストベクトルを印加すればよい．

一般に，k入力ANDゲートとNANDゲートのすべての縮退故障を検出するために，$k+1$個のテストベクトル$V1, V2, \ldots, Vk+1$が必要である．ここで，$Vi=<a1, a2, \ldots, ai\text{-}1, ai, ai+1, \ldots, ak+1>$, $ai=0, aj=1, j=1, 2, \ldots, k+1$, $i \neq j$である．また，k入力ORゲートとNORゲートのすべての縮退故障を検出するために，$k+1$個のテストベクトルNOT($V1$), NOT($V2$), ..., NOT($Vk+1$)が必要である．NOTの縮退故障を検出するには，0と1を印加すればよい．

2.3.1.3　スタック・オープン故障

CMOSゲートは，正常動作時にはp側とn側のどちらかが必ず導通状態にあるが，故障時にはどちらも非導通状態または導通状態になる可能性がある．つまり，CMOS論理回路ではゲートレベル回路の縮退故障モデルではモデル化できない故障が起こりうる [26, 44, 45].

例えば，図2.5に示したCMOSの2入力NANDゲートのトランジスタTaが永久開放状態になったとする．$<I1=0, I2=1>$を印加したとき，p側とn側のどちらも非導通状態になるので，出力は高抵抗状態になる．このとき，$<I1=0, I2=1>$を印加する直前の入力値により決まる出力値が保たれ，このときの出力値となる．つまり，このような故障により組合せ回路は順序回路のような動作をしてしまう．

このようにトランジスタの開放故障はゲートレベル回路の信号線の縮退故障モデルで表せないので，新しい故障モデルを設ける必要がある．この新しい故障モデルをスタック・オープン故障モデルといっている [26, 44].

縮退故障がゲートレベル回路の故障モデルであるのに対して，スタック・オープン故障モ

デルはトランジスタレベル回路の故障モデルである．1個の縮退故障を検出するためには，1 個のテストベクトルで十分なのに対して，1 個のスタック・オープン故障を検出するためには，連続する2個のテストベクトルが必要である [52]．例えば，図2.5の2入力NANDゲートのトランジスタ *Ta* のスタック・オープン故障を検出するには，まず初期化ベクトルと呼ばれる <*I*1 = 1, *I*2 = 1> を印加して，ゲートの出力値を 0 に設定する．続いてテストベクトルと呼ばれる <*I*1 = 0, *I*2 = 1> を印加する．正常時には，出力値は 1 であるが，故障時には，*p* 側と *n* 側のどちらも非導通状態になるので，出力値はその直前の出力値 0 となる．

テストベクトル対で複数のスタック・オープン故障を検出することができる場合もある．図 2.5 の 2 入力 NAND ゲートのトランジスタ *Tc* と *Td* のスタック・オープン故障を <*I*1 = *X*1, *I*2 = *X*2> と <*I*1 = 1, *I*2 = 1> というテストベクトルの対で検出することができる．ここで，*X*1 または *X*2 のいずれかが論理値 0 となる．2 入力 NAND ゲートのすべてのスタック・オープン故障を検出するベクトルの対は (<*I*1 = 1, *I*2 = 1>, <*I*1 = 0, *I*2 = 1>), (<*I*1 = 1, *I*2 = 1>, <*I*1 = 1, *I*2 = 0>), (<*I*1 = *X*1, *I*2 = *X*2>, <*I*1 = 1, *I*2 = 1>) である．ここで，*X*1 または *X*2 のいずれかが論理値 0 となる．

図 2.11 および図 2.12 に CMOS の *k* 入力 AND, OR, NAND, NOR ゲートの構成図を示す．*k* 入力 AND ゲートと NAND ゲートのすべてのスタック・オープン故障を検出するために，*k*+1 個のテストベクトル対 (*P*, *V*1), (*P*, *V*2), . . ., (*P*, *Vk*), (*Q*, *P*) が必要である．ここで，*P*=<1, 1, . . ., 1>, *Q*=<*x*1, *x*2, . . . , *xk*+1>, *x*1*x*2 . . *xk*+1 = 0, *Vi* = <*a*1, *a*2, . . ., *ai*-1, *ai*, *ai*+1, . . . , *ak*+1>, *ai* = 0, *aj* = 1, *j* = 1, 2, . . ., *k* + 1, *i* ≠ *j* である．また，*k* 入力 OR ゲートと NOR ゲートのすべてのスタック・オープン故障を検出するために，*k* + 1 個のテストベクトル対 (NOT(*P*), NOT(*V*1)), (NOT(*P*), NOT(*V*2)), . . ., (NOT(*P*), NOT(*Vk*)), (NOT(*Q*), NOT(*P*)) が必要である．NOT のスタック・オープン故障を検出するために，01 と 10 を印加すればよい．

2.3.1.4　単一故障と多重故障

一般に，単一故障とは回路に 1 個の故障しか起こらないと仮定する故障モデルであり，多重故障とは回路内に複数の故障が起こると仮定する故障モデルである．開発段階または製造の初期段階では，多重故障の可能性が高いが，使用段階では，単一故障を仮定しても十分である．また，上に述べたように，単一故障の検出を目標に生成されたテストベクトルにより多重故障が検出される場合もある．本論文では，主に単一縮退故障と単一スタック・オープン故障を対象故障とする．

2.3.2　故障検出と故障診断

2.3.2.1　基本用語

論理回路のテストを考察するとき，故障と誤りという概念を区別する必要がある．故障

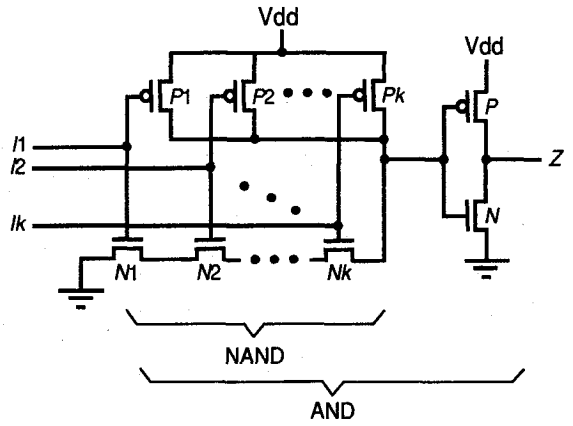


図 2.11　CMOS の k 入力 NAND と AND ゲート

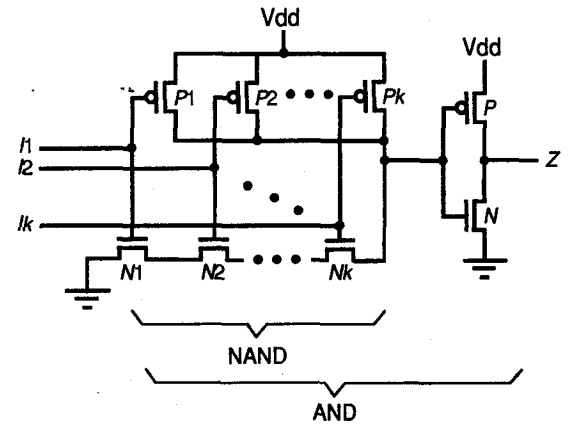


図 2.12　CMOS の k 入力 NOR と OR ゲート

(Fault) とは回路の内外に存在する回路動作を異常にさせる物理的な要素である．配線の断線や短絡，素子の物理的な欠陥は故障であるし，また，外部からの電磁波や α-線などによる影響も故障と見なされる．一方，誤り (Error) とは故障による回路動作に現れる異常である．ある信号波形が期待していたものと異なった波形を示したり，α-線の入射によりメモリの内容が変化することは誤りである．

テストベクトル，テスト系列およびテスト集合といった用語は頻繁に使用されている．テストベクトルとは同じ時刻にすべての入力線に印加される入力値であり，テスト系列とは1本の入力線に印加されるすべての入力値であり，テスト集合とはすべての入力線に印加されるすべての入力値である．これらの用語の区別を図 2.13 に示す．

2.3.2.2　故障検出

テスト生成を行なうにあたってまず，故障モデルを決めて対象故障リストを作る．それか

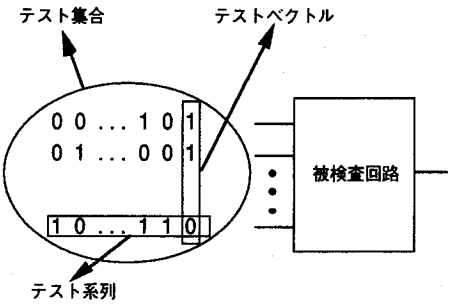


図 2.13　用語説明

ら，そのリストにあるすべての故障に対してテストベクトルを求める．

テスト生成を行なうとき，故障の影響が故障したゲートの出力線に現れるようにする故障設定および故障の影響が観測できる信号線に現れるようにする故障伝搬の両方を考慮する必要がある．外部出力端子しか観測できないテスト環境では，テスト生成は一般に困難であるが，全可観測な環境では，故障の影響を伝搬させる必要がなく，故障設定のみを行なえば充分ので，テスト生成は容易になる．その一方，全可観測な環境を実現するには，次章で述べるような特別な技術が必要なので，通常のテスト環境でテスト生成を行ない回路に故障があるかどうかを調べることが望まれる．

組合せ回路のテスト生成に関する研究は，従来から盛んに行なわれてきた．D [8], PODEM [9], FAN [10], CONT [11], そして SOCRATES [12] などのテスト生成のための優れたアルゴリズムが提案されている．これらの方法はいずれも縮退故障を故障モデルとしているが，最近，これらの方法に基づく，スタック・オープン故障検出用のテスト生成アルゴリズムも提案されている [26, 45, 52].

組合せ回路の場合に比べ，順序回路のテスト生成ははるかに困難である．順序回路のテスト生成手法は主に 2 種類ある．状態遷移表を用いる手法 [6, 16, 19] では，順序回路をブラックボックスと見なして，適当なテスト系列を印加し，状態遷移表で規定された回路の機能を確かめる．順序回路テスト問題を組合せ回路のテスト問題におきかえる手法 [3, 6, 16, 19] では，図 2.14 のように同期式順序回路を時間展開することにより，既存の組合せ回路のテスト生成アルゴリズムの利用を可能にする．

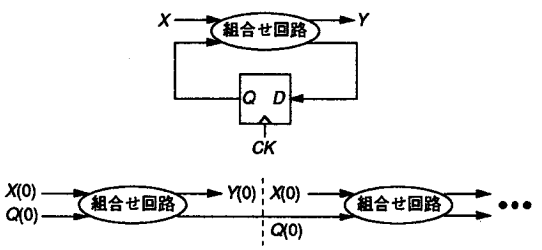


図 2.14　同期式順序回路の時間展開

2.3.2.3　故障診断

論理回路の故障診断とは故障箇所を見い出すことである．一般に故障診断を行なう前に，汎用テスタを用いて種々のテストベクトルによるテストや回路のブロック構成などを利用して，被疑範囲を可能な限り限定してから，信号線の観測により故障診断を行なう．現状では，被疑範囲を1Kゲート程度に絞ってから故障診断を行なうとの報告がある[31]．論理回路の故障診断には，以下の三つの手法がよく用いられている．

手法1（対話法）

この手法 [32] では，観測したい信号線を決め，そこに適当な論理値を設定して観測する．観測結果が期待値と一致するか否かにより故障の位置特定を行なう．□

この手法では，故障の状況，観測の結果から判断して，制御すべき信号線，論理値を決定する必要がある．したがってこの手法では，テスト実行者とのインタラクティブな対話を想定している．手法1では，人間の代わりにAI的な手法を駆使することも可能である[46]．しかし現状では，人間が観測の結果から判断を下し，故障を仮定し，故障診断を行なう方が効果的である．手法1の欠点は自動的に故障診断が行なえないことである．

手法2（ガイディド・プローブ法）

この手法[19,31]ではまず，故障モデルを仮定し，故障の影響が回路に現れるようにするテストベクトルを生成する．生成したテストベクトルを印加して回路に故障があるかどうかを調べる．回路の観測できる信号線で誤動作を観測したとき，故障は誤動作信号線の入力側にあるとして入力側の信号線を1本ずつプローブする．プローブとは，信号線の観測とそのシミュレーション値との比較照合である．入力側の信号線で分岐がある場合は，誤動作の存在する側にのみ遡る．以上の操作を故障ゲートが見つかるまで繰り返す．なお，故障ゲートとは入力線に誤動作がないのに出力線が誤動作しているゲートである．□

例えば，図2.15に示す回路の外部出力線で誤動作を観測したとする．このとき，ゲートAの入力線をプローブする．もしゲートCの出力線で誤動作を観測したら，次にゲートCの入力線をプローブする．もしゲートDの出力線で誤動作を観測したら，次にゲートDの入力線をプローブする．ゲートDの入力線に正常値を観測したら，ゲートDが故障していることが分かる．

この手法の長所はプローブすべき信号線のシミュレーションによる期待値さえあれば確実に故障ゲートを絞り込める点にある．したがって，この手法は自動的な故障診断システムに適用できる．その問題点は，段数の多い回路に対して，ゲート1段ずつ誤動作信号を追跡していたのでは，故障絞り込みの効率が悪いことである．またこの手法では，原則的にはす

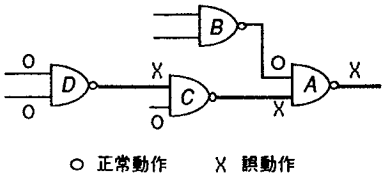


図 2.15　故障追跡の例

べての信号線に対する期待値が必要である．しかし，大規模な回路のすべての期待値を記憶していたのでは，その記憶容量が膨大になってしまう．

手法 3（故障辞書法）

これは内部信号線を含む観測できる信号線に現れる信号値の状況から故障箇所を推定する機能をもつテストベクトルを利用する方法である [19, 33]. □

故障辞書の例を図 2.16 に示す．ここで EB0001～EB0050 は内部信号線を含む観測できる信号線である．"TIME13705" などはテストベクトルを表す．"GOOD VECTOR" は正常時の出力値を示す．"－" は正常値であることを意味する．ここでは，縮退故障を対象にしている．例えば，テストベクトル "TIME 13705" を印加したときの観測結果により，故障番号（FAULT NUM）が 1676, 1681, 1684, 1689, 1692, 1697, 4252, 4254 である故障を診断することができる．なお，"ELEMENT" と "SIGNAL" はそれぞれゲートと信号線を意味する．

この手法は手法 2 の問題点を解決できる．故障辞書を利用すれば，故障の候補箇所を数箇所まで絞り込むことが可能なので，プローブすべき信号線数は少なくて済み，それらの期待値をすべて用意することも可能である．

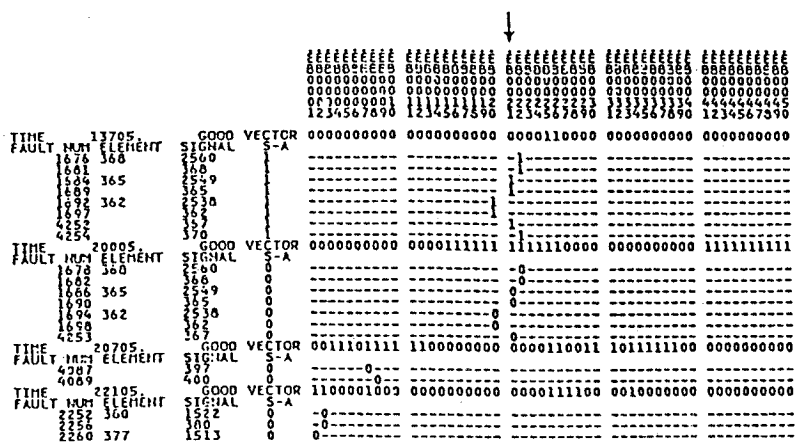


図 2.16　故障辞書の例

この手法には次の欠点がある．第1は，テスト生成が依然として困難で，テストベクトルと信号線の期待値の記憶容量が手法2のそれより少ないものの，大規模な回路の場合となると問題が残る．第2は，1個のテストベクトルを印加したあと，内部信号線を含むすべての観測点の論理値を測定する必要があって，時間がかかることである．

2.4　論理回路のテスト容易化設計

2.4.1　テスト容易化設計の必要性

組合せ回路のテスト生成問題はNP完全問題である [50, 51]．すなわち，その問題を入力数の多項式時間で解決するアルゴリズムを見つけることは極めて困難である．実際に使用される論理回路に対して効率的な生成手法はあるが，大規模な回路になると，生成時間が非常に長くなる場合がある．

順序回路のテスト生成問題は組合せ回路のそれよりはるかに困難である．状態遷移表を用いて順序回路のテスト生成を行なうとき，入力数が n，状態数が r の場合，テスト系列の長さは $O(n*r*r!)$ である [6]．このため，n と r が少しでも大きければ，この手法は現実的でなくなる．図2.14のように同期式順序回路を時間展開することにより，順序回路のテスト問題を組合せ回路のテスト問題におきかえる手法では，初期状態の設定，多重故障の取り扱い，入力数と素子数の増加などの問題がある．

論理回路のテストの本質的な困難さを原因に，高集積化と高機能化に伴い，論理回路のテスト生成時間が増大している．また，テストベクトル数も増大し，それを記憶するメモリ容量や印加時間も増大している．その結果，テストコストが全コストで占める割合いは年々増加する傾向にある [2-4]．

組合せ回路と順序回路のテスト生成問題を根本的に解決するために，設計段階においても，テストを考慮に入れる必要がある．すなわち，テスト容易な回路を設計することが必要である．テスト容易化設計の利点は，テスト生成が容易になり，テストの実施時間が短くなることである．その欠点は，素子数や外部入出力ピン数の増加，および動作速度の遅れなどである．そこで，総合的なコストの観点から，テスト容易化設計手法を使用するか，またどのテスト容易化設計手法をどの程度まで使用するかを判断する必要がある．

2.4.2　テスト容易化設計手法

2.4.2.1　概説

組合せ回路のテスト容易化設計手法は，三つのグループに分類することができる．まず，テスト生成を必要としない方法がある．その代表として，回路分割による全数テスト [15, 53] やシンドロームテスト容易化設計 [54] などがある．また，回路構造を限定することによ

りそのテストを簡単にする方法もある．このような方法では，半加算器アレー [55, 56]，PLA [57-59]，あるいはカットポイントセルラアレー [60] などの基本単位を用いて，規則性のある回路を構成する．さらに，回路変更による方法がある．その代表として，XOR ゲートを挿入する方法 [6] や回路を 2 入力ゲートで構成されるように変更する方法 [34, 35] などがある．

順序回路のテスト容易化設計手法は，状態図レベルのテスト容易化設計 [6] およびスキャン設計 [5-7, 18] に分類することができる．

次に，テスト容易化設計の代表例であるスキャン設計について説明する．

2.4.2.2　テスト容易化設計例ースキャン設計

図 2.17 に同期式順序回路の一例である．この回路の D- フリップ・フロップが並列ロードレジスタを構成していると見なすことができる．つまり，クロック・パルスにより，信号線 a, b の値が同時にフリップ・フロップにセットされる．順序回路のテストが困難である主な原因は，信号線 a', b' から組合せ部分に所要のテストベクトルを印加すること，および信号線 a, b の値を観測することが難しいためである．この考察をもとに提案されたのがスキャン設計である．スキャン設計にも様々な種類がある．以下では，スキャンパス設計について説明する．

スキャンパス設計では，D- フリップ・フロップをシフトレジスタラッチに改造する．図 2.18 にシフトレジスタラッチの記号を示す．$NT=0$ のとき，クロック・パルスにより D の値が Q にセットされる．$NT=1$ のとき，クロック・パルスにより S の値が Q にセットされる．クロック・パルスがないとき，Q の値が保持される．

このようなシフトレジスタラッチを図 2.19 に示すように接続する．このようにすれば，シフトレジスタラッチは 2 モードレジスタを構成することになる．このレジスタの動作モードは NT により選択される．正常動作 $(NT=0)$ のとき，従来の並列ロードレジスタとして動作し，テスト $(NT=1)$ のとき，シフトレジスタとして動作する．

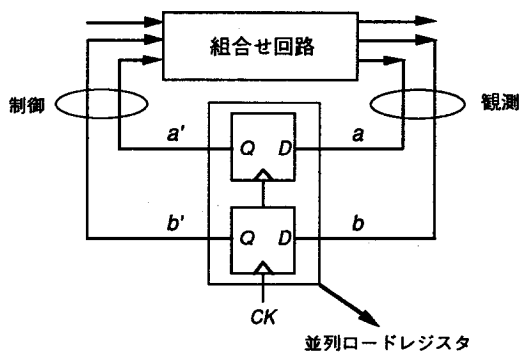


図 2.17　同期式順序回路の例

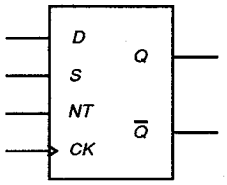


図 2.18　シフトレジスタラッチの記号

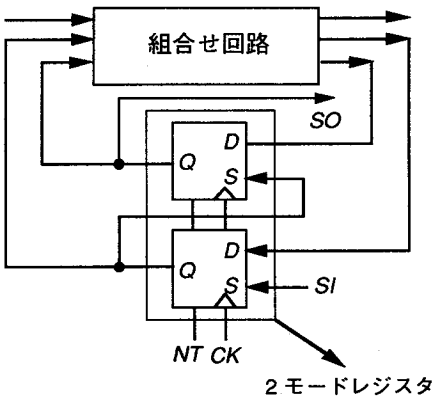


図 2.19　スキャンパス順序回路

スキャンパスが存在することにより，順序回路のテストの問題はその組合せ部分回路のテストの問題におきかえられる．例えば，図 2.20 (a) の回路の組合せ部分をテストするために，a, b, c から $x1, x2, x3$ を印加する必要がある．また，a', b', c' に現われる応答 $y1, y2, y3$ を観測する必要がある．スキャンパスがあれば，これらの操作は簡単にできる．

図 2.20 (b) に示すようにまずレジスタをテストモードにする．SI に $x2$ を与えて，クロック・パルスを印加すると，$x2$ を c にセットすることができる．次に図 2.20 (c) に示すように，SI に $x3$ を与えて，クロック・パルスを印加すると，$x2$ を b，$x3$ を c にセットすることができる．このとき，$x1$ を a に与えると，組合せ部分に所要のテストベクトルを印加することになる．

図 2.20 (c) に示したように，テストベクトル $x1, x2, x3$ に対する組合せ部分の応答を $y1, y2, y3$ とする．$y3$ は外部信号線 c' に現われているので簡単に観測できる．$y1, y2$ を観測するために，まずレジスタを正常モード，すなわち並列ロードモードにする．このとき，クロック・パルスを印加すると，図 2.20 (d) に示すように，$y1, y2$ は a, b にセットされる．これで，SO から $y1$ を観測することができる．次に，レジスタをテストモードにしてクロック・パルスを印加すると，図 2.20 (e) に示すように，$y2$ は b にシフトされる．つまり，SO から $y1$ を観測することができる．

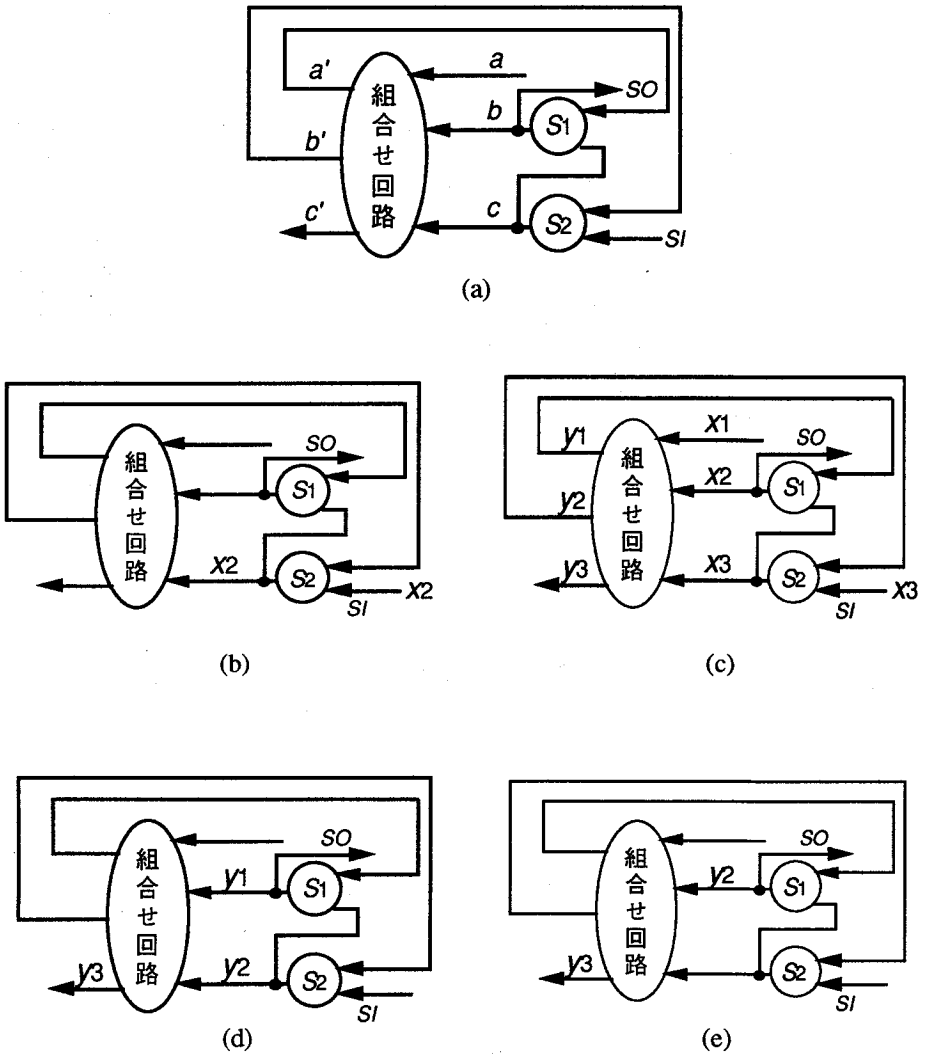


図 2.20　スキャンパス順序回路のテスト例

この例から明らかなように，順序回路のテストの問題は，スキャンパスを利用することにより，その組合せ部分回路のテストの問題におきかえられ，現実的に解決できる問題となる．

2.4.2.3　組込み自己テスト（BIST）の概念

実際のテストは通常テスタと呼ばれる高価な設備でを行なわれる．テスタでは，テストベクトルの生成，記憶，印加およびテスト結果の判定などが行なわれる．テストベクトル数が多くなると，高価なテスタを専用する時間が長くなり，テストコストも当然高くなる．テス

ト生成のコストのみではなく，テスト実施のコストをも低減させるために，組込み自己テスト（BIST）の概念が提案されている [5-7, 17, 19].

図 2.21 に示すように，組込み自己テストでは，テストベクトルの生成と印加，テスト結果の判定などのテスタ機能を回路内部で実現することにより，テスト実施のコストを安くする．組込み自己テストでは，テスト生成部を内蔵するので，ハード的に簡単に実現できるものでなければならない．そこで，疑似ランダム入力，全数入力や規則的に発生される入力などを用いるテストが多用されている．また，被テスト回路の応答をそのまま期待値と比較するのではなく，一旦圧縮されてから比較を行なう．この圧縮によく用いられるのは，カウンタや LFSR などである．

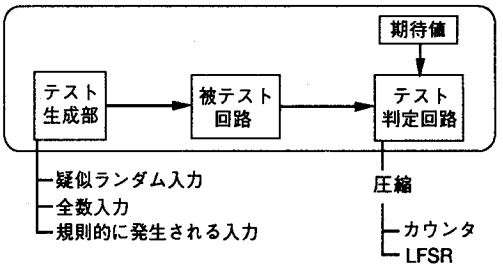


図 2.21　BIST の概念

2.5　まとめ

本章ではまず，本研究で対象回路とする組合せ回路と順序回路の概念について説明した．特に論理回路の基本論理ゲートおよび D-フリップ・フロップによる表現法について説明した．順序回路は構造的に組合せ回路より複雑で，そのテストもはるかに困難である．

次に，論理回路のテストの基本概念，特に縮退故障やスタック・オープン故障などの重要な故障モデルについて説明し，また k 入力 AND, OR, NAND, NOR ゲートおよび NOT ゲートのすべての縮退故障やスタック・オープン故障を検出するテストベクトルを示した．

さらに，論理回路のテスト容易化設計の基本概念および主な手法について紹介した．特に順序回路のスキャンパス設計について詳しく説明した．スキャンパス設計では，回路内のすべてのフリップ・フロップはテスト時にシフトレジスタを形成するため，回路へのテストベクトルの印加および回路内部状態の観測は大幅に簡単化される．

3 全可観測な環境

全可観測な環境は，電子ビームプロービング技術の使用を背景としている．本章では，電子ビームプロービング技術について述べ，全可観測な環境でのテストの特徴と全可観測な環境におけるテスト容易化設計の必要性について説明する．

3.1　電子ビームプロービング技術

電子ビームプロービング技術の母体は走査形電子顕微鏡である [47]．動作状態の集積回路チップ表面を電子ビーム走査したとき検出される 2 次電子の量は表面に形成される電界分布により影響を受ける．一般に，電圧の低い部分は検出された2次電子が多いため明るく，電圧の高い部分は検出された 2 次電子が少ないため暗いという電圧コントラスト像が得られる [23]．また 2 次電子のエネルギー分布は信号線の電圧だけ移動するので，エネルギー分析器により信号線の電圧を定量的に測定することができる [23]．一方，波形を測定する場合には，検出系の応答速度に限界があるため，ストロボ法を用いる [27]．すなわち，図3.1に示すように，電子ビームを短いパルスにし，その周期を被観察回路の動作周期と一致させて信号線を照射する．電子ビームと回路動作の位相を少しずつ動かして，2次電子を検出することにより波形の観測が可能となる．また図 3.2 に示すように，位相を固定したまま電子ビームを 2 次元走査することにより，特定の位相状態での電圧像が得られる．

この原理に基づき，集積回路の故障診断を行なう装置は，一般に電子ビームテスタと呼ばれている．従来の CAD 環境との統合により，観測したい信号線を論理図で指定するだけで，その信号線へのビームの位置決め，プロービング，結果の処理と表示が自動的に行なわれる高性能な電子ビームテスタが開発されている [30].

電子ビームテスタの構成や機能は，その目的とする診断内容や適用デバイスによりそれぞれ若干異なる．一般の電子ビームテスタは，電子銃から発生した電子ビームを細く絞って集積回路チップ表面の測定点に照射する電子ビーム鏡筒，電子ビームを短いパルスにするパルスゲート，電子レンズ，偏向器，被観察回路を挿入する真空の試料室などを含む．真空の試料室にはモータ駆動のステージ，エネルギー分析器，2次電子検知器などが含まれている．

さらに 2 次電子信号の処理回路，制御装置，ストロボ回路，被観察回路のトライバ，像と波形の表示部，計算機などから構成されている．被観察回路の損傷を避けるため電子ビームの加速電圧は 1 Kv 前後と低い．

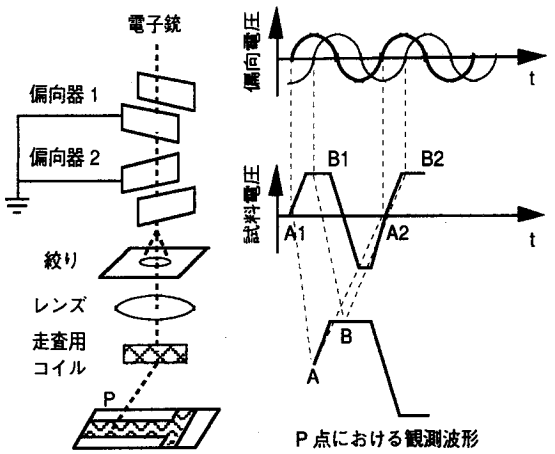


図 3.1　ストロボ法による電圧波形測定

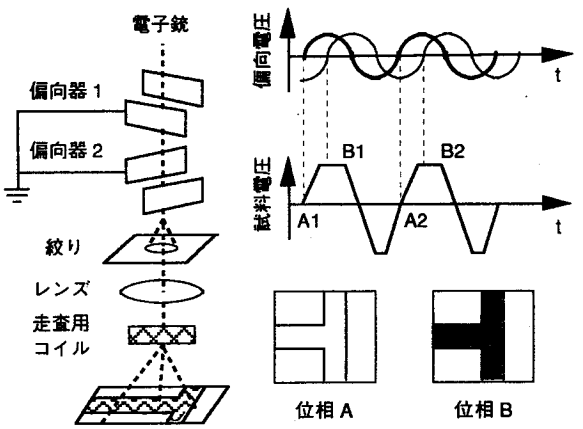


図 3.2　ストロボ法による電圧像測定

電子ビームテスタを用いる場合，集積回路チップの最上位層の信号線を直接に観測できる．報告によると，2 層 Al 配線の回路なら 95% の信号線を観測できる電子ビームテスタが既に開発されている．特に，ゲートアレイの場合，ゲート間の信号線が最後に形成されるので，大部分のゲートの出力線が最上位層の信号線として現われる．表3.1 はいくつかの実際の回路の最上位層への配線率を示す．

表3.1　最上位層への配線率

回路	ゲート数	配線数	最上位層配線率
A	2.3 K	303	96.7 %
B	3.5 K	2643	99.1 %
C	15.0 K	6331	98.4 %
D	15.5 K	5851	99.4 %

電子ビームプロービング技術の特徴は以下の通りである．

(1)　機械的に非接触，非破壊である．

(2)　スポットが微小である．

(3)　配線との間に容量がなく高速の波形観測が可能である．

(4)　ビームの位置決めが容易である．

3.2　全可観測な環境

電子ビームテスタを用いて回路内部の多くの信号線の論理値を観測することができるので，テストを行なうとき，故障の影響を外部出力線まで観測のために伝搬させる必要がない．したがって，ゲートレベルの回路を対象にするとき，回路内の各ゲートの出力線が観測できると仮定することは非現実ではない．このようなテスト環境を全可観測なテスト環境，あるいは単に全可観測な環境と呼ぶ．これに対し，外部出力線しか観測できないテスト環境を通常のテスト環境と呼ぶ．

3.3　全可観測な環境でのテストとテスト容易化設計

全可観測な環境でテストを行なうとき，故障の影響を観測できる信号線まで伝搬させる必要がない．したがって，全可観測な環境において，テスト生成は簡単にできる．

しかし，全可観測な環境で使用できるテスト系列はいくつかの要求を満たさなければならない．これについて第1章でも触れたが，基本的にはテスト系列が短くなければならない．また，電子ビームテスタなどを用いるとき，その短いテスト系列を回路に繰り返し印加しなければならない．全可観測な環境で生成されたテスト系列はどの回路の場合でも十分短いという保証がない．また，順序回路でよくあるように，その組合せ部分が短いテスト系列を有しても，それを連続的に回路に繰り返し印加することができない．

全可観測な環境での中心問題は，如何にして短いテスト系列を作るか，および如何にしてそれを連続的に回路に繰り返し印加するかである．これまでは，人手によりなるべく短いテスト系列を作ることが考えられていた．これは技術者の経験によるところが大きく，費用も

時間もかかる場合が多い．自動的に生成を行なう試みもあったが[29, 32]，どのような回路に対しても有効であるという保証はない．このように回路構造を変えずにアルゴリズムを改良するだけでは自ら限界がある．この問題を根本的に解決するためにテスト容易化設計を考える必要がある．

全可観測な環境の概念やそれを実現する技術の確立は近年のことであるが，かつて，最小長のテスト系列でテスト可能な回路をどのように構成すればよいかなどを調べる研究が行われていた[34, 35]．それは，テスト系列の長さの下限を知っていれば，それより短いテスト系列で回路をテストしようとする無駄な努力が省かれるからであった．また，テスト系列の長さの下限は実際に生成したテスト系列の長さの評価の尺度にもなるからであった．次に，その代表的な研究例[35]を紹介する．

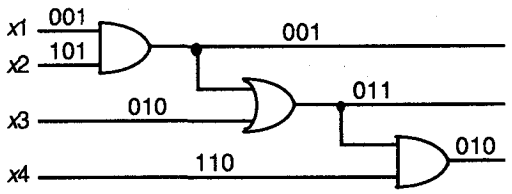


図3.3　3入力でテスト可能な回路例

簡単な例として図3.3の回路について説明する．2入力ANDゲートの縮退故障は$SA = \{$ 011, 101, 110 $\}$に属する2個の系列でテスト可能であり，2入力OR素子の縮退故障は$SO = \{$ 100, 010, 001 $\}$に属する2個の系列でテスト可能である．したがって，図3.3のように入力割当を行なえば，長さ3の系列ですべての縮退故障がテスト可能となる．

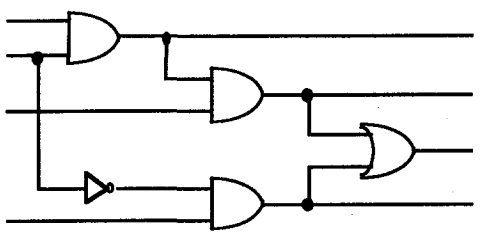


図3.4　3入力でテスト可能でない回路例

しかし，図3.4の回路の場合には3入力でテスト可能ではない．このとき，次のような方法で3入力でテスト可能なように変換することができる．

(1)　与えられた組合せ回路を2入力のANDゲートとORゲートとNOTゲートを用いて樹枝状回路として実現する．

(2)　外部入力線に *SA* または *SO* の系列を割り当てる．各ゲートについて割り当てられた入力系列がテスト可能ならばその系列を外部出力線に伝搬させる．テスト可能でなければ，AND または OR ゲートを用いて展開し，テスト可能なようにする．

(3)　すべてのゲートに系列が割り当てられるまで (2) を繰り返す．

例えば，図 3.4 の回路は図 3.5 の回路のように 3 入力テスト可能とすることができる．このとき，2 個のゲートと 2 本の外部入力線，2 本の外部出力線が付加されている．テスト系列は，その長さを 3 より短くすることができないので，これは最小長のテスト系列となる．

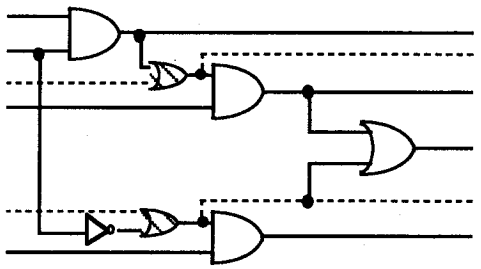


図 3.5　3 入力でテスト可能な回路への変換

しかし，このテスト容易化設計には以下のような欠点がある．

(1)　対象故障は信号線の縮退故障のみである．現在 CMOS 技術が多用されているため，CMOS に特有のスタック・オープン故障をも対象にする必要がある．

(2)　回路を構成する素子の種類や入力数についての制限が厳しい．

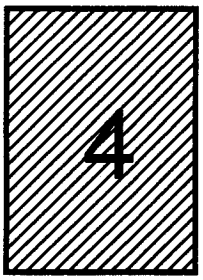
(3)　回路変換は主に置換で行なわれ，オーバーヘッドが大きいと思われる．

(4)　故障診断手法が示されていない．

そこで，これからの章では，より広範的，系統的なアプローチを取り，テスト容易な論理回路の概念，テスト方法，および構成方法について提案する．

3.4 まとめ

電子ビームプロービング技術の基本原理について述べた．電子ビームプロービング技術を用いることにより，回路内の最上位層にある信号線を観測することができる．この技術を背景に，回路のすべてのゲートの出力線が観測できるテスト環境，いわゆる全可観測な環境を定義した．電子ビームプロービング技術を使用する前提条件として，被観測回路が短いテスト系列をもち，かつその系列が回路に繰り返し印加することができることである．このため，全可観測な環境においてもテスト容易化設計が必要である．



全可観測な環境での組合せ回路のテスト

本章では，全可観測な環境でのテスト容易な組合せ回路について述べる [36-40]．基本的な考え方は回路構造を限定することである．本章ではまず，k-UCP 回路の定義，テスト生成法，および故障検査法について述べる．次に，k-UCP 回路の拡張である k-R 回路の概念について述べる．

4.1　k-UCP 回路の定義

定義 4.1　k 入力 AND, OR, NAND, NOR ゲートおよび NOT ゲートで構成される組合せ回路を，k-U 回路と呼ぶ．

定義 4.2　k-U 回路において，k 入力のゲートの入出力線に異なる色を，NOT ゲートの入出力線とファンアウトの信号線に同じ色を塗るように，$k+1$ 種の色ですべての信号線を塗ることができるとき，この k-U 回路は $k+1$ 色解をもつという．$k+1$ 色解をもつ k-U 回路を k-UC 回路と呼ぶ．

定義 4.3　k-U 回路において，表 4.1 に示すルールにしたがって，正極性（+）と負極性（−）をすべての信号線に割り当てることができるとき，この k-U 回路は正しい極性をもつという．正しい極性をもつ k-U 回路を k-UP 回路と呼ぶ．

定義 4.4　$k+1$ 色解と正しい極性をもつ k-U 回路を k-UCP 回路と呼ぶ．

図 4.1 と図 4.2 にそれぞれ 2-UCP 回路と 3-UCP 回路の例を示す．一般に，任意の組合せ回路は付加ゲートにより k-UCP 回路に変換できる．図 4.3 にその一例を示す．図 4.3 (a) の回路の外部入力線にどのような 3 色の色塗りをしてもゲート E の入力線の色が必ず同じになるので，図 4.3 (a) の回路は k-UCP 回路ではない．この回路は 2 個の NAND ゲートを

表 4.1　k - UCP 回路の極性割当ルール

	入力	出力
NAND	+	+
AND	+	-
NOR	-	-
OR	-	+
NOT	-/+	+/-

付加することにより図 4.3 (b) の 2 - UCP 回路に変換できる．回路変換手法については，次章で詳しく述べる．

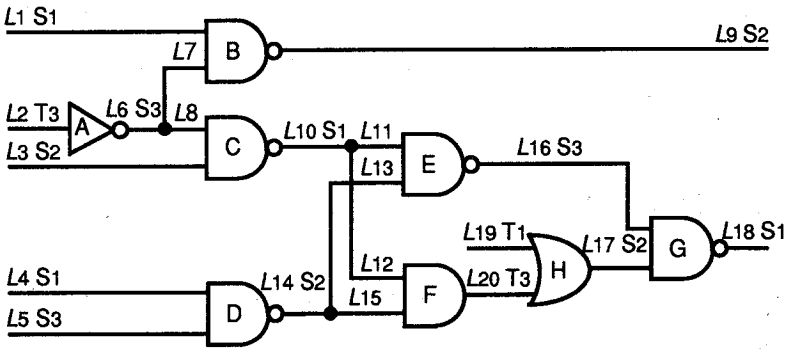


図 4.1　2 - UCP 回路の例

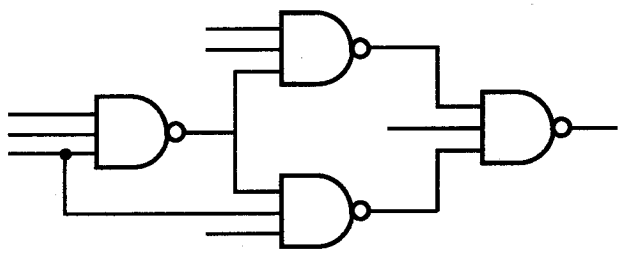
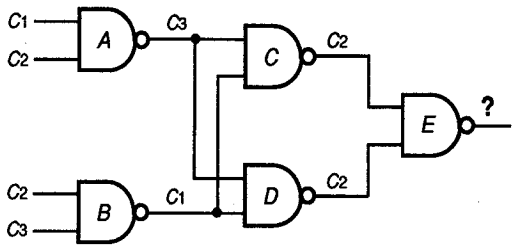


図 4.2　3 - UCP 回路の例

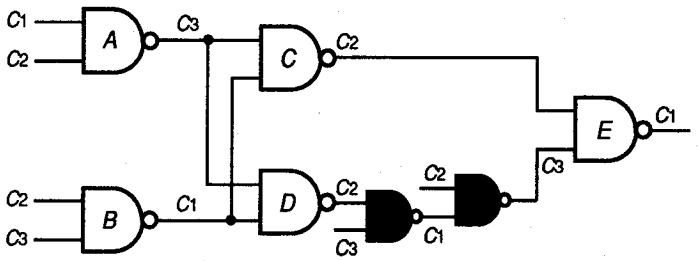
k - UCP 回路を構成するゲートの種類をさらに限定することにより，k - UC - NAND 回路と k - UC - NOR 回路が得られる．

定義 4.5　k 入力 NAND ゲートからなる k - UCP 回路を k - UC - NAND 回路と呼ぶ．

定義 4.6　k 入力 NOR ゲートからなる k - UCP 回路を k - UC - NOR 回路と呼ぶ．



(a)　k-UCP でない回路例



(b)　2-UCP 回路への変換

図 4.3　回路変換の例

明らかに，NAND ゲートまたは NOR ゲートからなる組合せ回路は必ず正しい極性をもつ．

4.2　基本系列

k-UCP 回路のテスト集合は，あらかじめ生成されている $2(k+1)$ 個の系列から，与えられた k-UCP 回路に応じて作られる．この $2(k+1)$ 個の系列を k-基本系列と呼ぶ．k-UCP 回路の k-基本系列は回路構造に関係なく，故障モデルと k の値により決まる．以下では，スタック・オープン故障用と縮退故障用の k-基本系列の定義と生成法について述べる．

4.2.1　基本系列の定義

定義 4.7　スタック・オープン故障用の k-基本系列は，次の条件を満たす $2(k+1)$ 個の論理値 0 と 1 からなる系列 $S1, S2, \ldots, Sk+1, T1, T2, \ldots, Tk+1$ である．

(1)　$\mathrm{NAND}(S1, S2, \ldots, Si-1, Si+1, \ldots, Sk+1) = Si$

$\mathrm{NOT}(Si) = Ti$

(2)　$S1, S2, \ldots, Si-1, Si+1, \ldots, Sk+1$ は k 入力 NAND ゲートのすべてのスタック・オープン故障を検出することができ，Si は NOT ゲートのすべてのスタック・オープン

故障を検出することができる.

補題 4.1　スタック・オープン故障用の k-基本系列 $S1, S2, \ldots, Sk+1, T1, T2, \ldots, Tk+1$ は次の性質をもつ.

(1) $\mathrm{AND}(S1, S2, \ldots, Si-1, Si+1, \ldots, Sk+1) = Ti$
$\mathrm{OR}(T1, T2, \ldots, Ti-1, Ti+1, \ldots, Tk+1) = Si$
$\mathrm{NAND}(S1, S2, \ldots, Si-1, Si+1, \ldots, Sk+1) = Si$
$\mathrm{NOR}(T1, T2, \ldots, Ti-1, Ti+1, \ldots, Tk+1) = Ti$
$\mathrm{NOT}(Si) = Ti, \mathrm{NOT}(Ti) = Si$

(2) $S1, S2, \ldots, Sk+1$ ($T1, T2, \ldots, Tk+1$) は k 入力 AND と NAND (OR と NOR) ゲートのすべてのスタック・オープン故障を検出することができ，Si と Ti は NOT ゲートのすべてのスタック・オープン故障を検出することができる.

次に示す系列はスタック・オープン故障用の 2-基本系列である.

$$S1 = 1\,0\,1\,0\,1\,1\,1 \qquad T1 = 0\,1\,0\,1\,0\,0\,0 \qquad (4.1.a)$$
$$S2 = 1\,1\,0\,1\,1\,0\,1 \qquad T2 = 0\,0\,1\,0\,0\,1\,0 \qquad (4.1.b)$$
$$S3 = 0\,1\,1\,1\,0\,1\,0 \qquad T3 = 1\,0\,0\,0\,1\,0\,1 \qquad (4.1.c)$$

定義 4.8　縮退故障用の k-基本系列は，次の条件を満たす $2(k+1)$ 個の論理値 0 と 1 からなる系列 $S1, S2, \ldots, Sk+1, T1, T2, \ldots, Tk+1$ である.

(1) $\mathrm{NAND}(S1, S2, \ldots, Si-1, Si+1, \ldots, Sk+1) = Si$
$\mathrm{NOT}(Si) = Ti$

(2) $S1, S2, \ldots, Si-1, Si+1, \ldots, Sk+1$ は k 入力 NAND ゲートのすべての縮退故障を検出することができ，Si は NOT ゲートのすべての縮退故障を検出することができる.

補題 4.2　縮退故障用の k-基本系列 $S1, S2, \ldots, Sk+1, T1, T2, \ldots, Tk+1$ は次の性質をもつ.

(1) $\mathrm{AND}(S1, S2, \ldots, Si-1, Si+1, \ldots, Sk+1) = Ti$
$\mathrm{OR}(T1, T2, \ldots, Ti-1, Ti+1, \ldots, Tk+1) = Si$
$\mathrm{NAND}(S1, S2, \ldots, Si-1, Si+1, \ldots, Sk+1) = Si$
$\mathrm{NOR}(T1, T2, \ldots, Ti-1, Ti+1, \ldots, Tk+1) = Ti$
$\mathrm{NOT}(Si) = Ti, \mathrm{NOT}(Ti) = Si$

(2) $S1, S2, \ldots, Sk+1$ ($T1, T2, \ldots, Tk+1$) は k 入力 AND と NAND (OR と NOR) ゲートのすべての縮退故障を検出することができ，Si と Ti は NOT ゲートのすべての縮退

故障を検出することができる．

次に示す系列は縮退故障用の2-基本系列である．

$S1 = 0\,1\,1$　　$T1 = 1\,0\,0$　　(4.2.a)

$S2 = 1\,0\,1$　　$T2 = 0\,1\,0$　　(4.2.b)

$S3 = 1\,1\,0$　　$T3 = 0\,0\,1$　　(4.3.c)

明らかに，ある故障モデルに対する k-基本系列は唯一ではない．例えば，4.1式に示したスタック・オープン故障用の2-基本系列は，縮退故障用の2-基本系列でもある．

4.2.2　基本系列の生成

4.2.2.1　スタック・オープン故障用の基本系列の生成

スタック・オープン故障用の k-基本系列は次の手続きにより生成される．

手続き BG1（スタック・オープン故障用の k-基本系列を生成する）

(1)　$k(k+1)$ 個の対 $<i, j>$ $(i, j = 1, 2, \ldots, k+1; i \neq j)$ に番号 $1, 2, \ldots, k(k+1)$ を次のように割り当てる．

(1-a)　対 $<k+1, 1>$ に番号1を割り当てる．$m = 1$ とする．

(1-b)　これまで割り当てられた番号の最大値を m とし，m をもつ対を $<h, i>$ とする．

(1-c)　番号が割り当てられておらずかつ1番目の要素が i である対の集合 $SP = \{<i, \triangleright>\}$ を求める．

(1-d)　SP が空であれば，(2)へ移る．そうでなければ，SP の中で一番小さい t をもつ対に番号 $m+1$ を割り当て，(1-b)に戻る．

(2)　(1)で番号が割り当てられた $k(k+1)$ 個の対 $<i, j>$ $(i, j = 1, 2, \ldots, k+1; i \neq j)$ を番号順に並べて，系列 R: $<k+1, 1>, <1, 2>, \ldots, a, b, c, \ldots$ を形成する．

(3)　系列 R を構成する対の2番目の要素を取り出し，一番前に $k+1$ を加えて，系列 S: $k+1, 1, 2, \ldots, a, b, c, \ldots$ を形成する．

(4)　$Pt = (x1, x2, \ldots, xt-1, xt, xt+1, \ldots, xk+1)'$ とする．ここで，$xt = 0$, $xi = 1$, $i = 1, 2, \ldots, k+1$, $i \neq t$ とする．$P1, P2, \ldots, Pk+1$ を系列 S の示す順番に並べると，系列 P: $Pk+1, P1, P2, \ldots, Pa, Pb, Pc, \ldots$ が得られる．

(5)　系列 P を論理値0と1からなる行列と見なす．この行列の i 行目を Si と定義する．また，Ti を Si の否定と定義する．なお，$i = 1, 2, \ldots, k+1$ とする．　□

補題 4.3　手続き BG1 の (1) では，すべての対に番号が割り当てられる．また，番号 g をもつ対の2番目の要素と番号 $g+1$ をもつ対の1番目の要素は等しい（$g=1,2,\ldots,k(k+1)-1$）．

（証明）　$k=2$ と $k=3$ のとき，この補題の前半が明らかに成立する．$k=t$ のとき，この補題の前半が成立すると仮定する．$k=t+1$ のときの対を図4.4に示す．対 $<t+2,2>$ に番号が割り当てられるまで，対 $<t+2,1>$ を含む行と対 $<1,2>$ を含む列にあるすべての対に番号が割り当てられている．番号が割り当てられていない対 $<i,j>$ を $<i-1,j-1>$ に書き換えると，$k=t$ のときの対となる．よって，この補題の前半が成立する．この補題の後半が成立することは手続き BG1 の (1) より明かである．　□

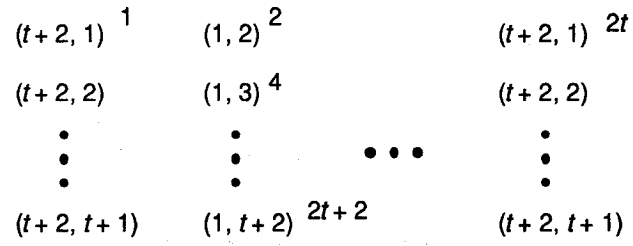


図 4.4　$k=t+1$ のときの対の番号付け

補題 4.4　系列 P には，Pi に Pj が続く対が1個しかない（$i,j=1,2,\ldots,k+1; i\neq j$）．

（証明）　任意の i,j（$i,j=1,2,\ldots,k+1; i\neq j$）に対して，系列 S には i に j が続く対が1個しかないことを証明すればよい．系列 R に対 $<i,j>$ があるので，i に j が続く対は必ず系列 S に存在する．また，もし系列 S には i に j が続く対が存在すれば，系列 R には対 $<i,j>$ が必ず存在する．系列 R に対 $<i,j>$ が1個しかないので，系列 S にも i に j が続く対が1個しかないことが分かる．よって，この補題が成り立つ．　□

定理 4.1　手続き BG1 で求められた $2(k+1)$ 個の系列 $S1, S2, \ldots, Sk+1, T1, T2, \ldots, Tk+1$ はスタック・オープン故障用の k-基本系列である．

（証明）　Ti の定義より，$\mathrm{NOT}(Si)=Ti$ である．次に，$\mathrm{NAND}(S1, S2, \ldots, Si-1, Si+1, \ldots, Sk+1)=Ti$ が成立することを証明する．なお，$i=1,2,\ldots,k+1$ とする．

Si の t ビット目の値を $Si(t)$ とする．Pi の定義より，$S1, S2, \ldots, Sk+1$ には0が1個しかない．それで，任意の t について，$\mathrm{NAND}(S1(t), \ldots, Si-1(t), Si+1(t), \ldots, Sk+1(t))=Si(t)$ である．よって，$\mathrm{NAND}(S1, S2, \ldots, Si-1, Si+1, \ldots, Sk+1)=Si$ が成り立つ．

系列 P に $Pi\,P1, \ldots, Pi\,Pi-1, Pi\,Pi+1, \ldots, Pi\,Pk$ および $Pj\,Pi$（$i\neq j$）があるので，CMOS の k 入力 NAND ゲートのすべての単一スタック・オープン故障を検出することができるテストベクトルが $S1, S2, \ldots, Si-1, Si+1, \ldots, Sk+1$ に含まれていることが分かる．　□

明らかに，手続き BG1 で求められたスタック・オープン故障用の k - 基本系列の長さは $k(k+1)+1$ である．

Si には 0 と 1 が含まれているので，それでNOTゲートのすべての縮退故障を検出することができる．また，$S1, S2, \ldots, Si-1, Si+1, \ldots, Sk+1$ には，<0, 1, . . ., 1>, <1, 0, . . ., 1>, . . ., <1, 1, . . ., 0>, <1, 1, . . ., 1> が含まれている．2.3.1.2 で述べたように，これらの系列は k 入力 NAND ゲートのすべての縮退故障を検出することができる．よって，手続き BG1 で求められた $2(k+1)$ 個の系列 $S1, S2, \ldots, Sk+1, T1, T2, \ldots, Tk+1$ は縮退故障用の k - 基本系列でもある．

例 4．1　手続き BG1 を用いて 3 - UCP 回路のスタック・オープン故障用の基本系列を生成する場合，各段階での実行結果は次の通りである．

(1) では，12 個の対 $<i,j>$ $(i,j=1,2,3; i \neq j)$ に番号 1, 2, . . ., 12 を割り当てる．その結果は次に示す．

<4，1>1	<1，2>2	<2，1>3	<3，1>5
<4，2>7	<1，3>4	<2，3>8	<3，2>9
<4，3>11	<1，4>6	<(2，4>10	<3，4>12

(2)で得られた系列 R は <4, 1>, <1, 2>, <2, 1>, <1, 3>, <3, 1>, <1, 4>, <4, 2>, <2, 3>, <3, 2>, <2, 4>, <4, 3>, <3, 4> である．

(3) で得られた系列 S は 4, 1, 2, 1, 3, 1, 4, 2, 3, 2, 4, 3, 4 である．

(4) $P1 = (0, 1, 1, 1)'$, $P2 = (1, 0, 1, 1)'$, $P3 = (1, 1, 0, 1)'$, $P4 = (1, 1, 1, 0)'$ とする．得られた系列 P は $P4, P1, P2, P1, P3, P1, P4, P2, P3, P2, P4, P3, P4$ である．

(5) 系列 P を次のような論理値 0 と 1 からなる行列と見なす．

1 0 1 0 1 0 1 1 1 1 1 1 1
1 1 0 1 1 1 1 0 1 0 1 1 1
1 1 1 1 0 1 1 1 0 1 1 0 1
0 1 1 1 1 1 0 1 1 1 0 1 0

よって，得られた基本系列は次の通りである．

$S1$ = 1 0 1 0 1 0 1 1 1 1 1 1 1	$T1$ = 0 1 0 1 0 1 0 0 0 0 0 0 0
$S2$ = 1 1 0 1 1 1 1 0 1 0 1 1 1	$T2$ = 0 0 0 0 1 0 0 0 1 0 0 1 0
$S3$ = 1 1 1 1 0 1 1 1 0 1 1 0 1	$T3$ = 0 0 1 0 0 0 0 1 0 1 0 0 0
$S4$ = 0 1 1 1 1 1 0 1 1 1 0 1 0	$T4$ = 1 0 0 0 0 0 1 0 0 0 1 0 1

4.2.2.2　縮退故障用の基本系列の生成

縮退故障用の k-基本系列は次の手続きにより生成される．

手続き BG2（縮退故障用の k-基本系列を生成する）

(1)　$i=1$ とする．

(2)　$Si=a1\ a2 \ldots ak+1$ とする．なお，$ai=0, aj=1, j \neq i, j=1,2,\ldots,k+1$．$Ti=\mathrm{NOT}(Si)$ とする．

(3)　$i=i+1$ とする．$i \leqq k+1$ であれば，(2) に戻る．　□

手続き BG2 より，次の定理は明らかである．

定理 4.2　手続き BG2 で求められた $2(k+1)$ 個の系列 $S1, S2, \ldots, Sk+1, T1, T2, \ldots, Tk+1$ は縮退故障用の k-基本系列である．

手続き BG2 で求められた縮退故障用の k-基本系列の長さは $k+1$ である．手続き BG1 で求められた基本系列と異なって，手続き BG2 で求められた基本系列は縮退故障にのみ有効である．

4.3　k-UCP 回路の故障検出

k-UCP 回路のテスト集合は次の手続きにより簡単に生成できる．

手続き TG（k-UCP 回路のテスト集合を生成する）

(1)　故障モデルと k の値に応じて，手続き BG1 または手続き BG2 で k-UCP 回路の $2(k+1)$ 個の k-基本系列を求める．

(2)　与えられた k-UCP 回路の信号線に塗られた $k+1$ 種の色を $C1, C2, \ldots, Ck+1$ とする．色 Ci と正極性をもつ外部入力線に k-基本系列の Si を，色 Ci と負極性をもつ外部入力線に k-基本系列の Si を対応させる．　□

手続き TG で外部入力線に対応させられた基本系列はその回路のテスト集合となる．

定理 4.3　全可観測な環境で，k-UCP 回路のすべての単一縮退故障は長さ $k+1$ のテスト系列で検出でき，単一縮退故障と単一スタック・オープン故障は長さ $k(k+1)+1$ のテスト系列で検出できる．

（証明）　k-UCP 回路の定義と基本系列の性質（補題4.1と補題4.2）より分かるように，手続き TG で求められたテスト集合を k-UCP 回路に印加したとき，回路内の任意のゲー

トの入力線に，その信号線が色 Ci と正極性をもつ場合に Si，その信号線が色 Ci と負極性をもつ場合に Ti が印加されることになる．補題4.1と補題4.2より，全可観測な環境では，k-UCP回路のすべての単一縮退故障と単一スタック・オープン故障が検出できる．　□

例 4.2　図4.1の2-UCP回路のテスト集合を求める．まず，図4.1の回路のすべての信号線に色を塗り，極性を割り当てる．$C1, C2, C3$ は3種類の色を表す．手続き TG で求められたテスト集合は次の通りである．

$$S1 \rightarrow L1, L4;\ S2 \rightarrow L2;\ S3 \rightarrow L5;\ T3 \rightarrow L2.$$　□

k-UCP回路のテスト集合の生成は極めて簡単で，所要時間も少ない．また，実用的な k の値が2または3であるので，テスト系列の長さは7または13と短い．このため，テスト集合を記憶するためのメモリ容量は少ない．さらに，シミュレーションを行なわなくても，各信号線の色と極性よりその信号線のテスト集合に対する期待値を得ることができ，これらの期待値を保存する必要がない．したがって，k-UCP回路の使用は，集積回路の大規模化によるテスト集合の生成時間，シミュレーションの時間，およびテスト系列と期待値の記憶容量の増加問題の解決策となる．

4.4　通常のテスト環境における k-UCP 回路の故障検出

このように，全可観測な環境において k-UCP 回路の故障検出は少ないテストベクトルで行なえることを明らかにした．本節では，それらのテストベクトルの，通常のテスト環境における故障検出率を求める．これは，k-UCP回路に適した故障シミュレーション手法により行なわれる．

以下では，説明を簡単にするため，2-UC-NAND 回路の場合のみについて述べる．図4.5に2-UC-NAND回路の例を示す．R, G, Y は3種類の色を示す．得られる結論は一般の k-UCP 回路の場合にも簡単に拡張できる．

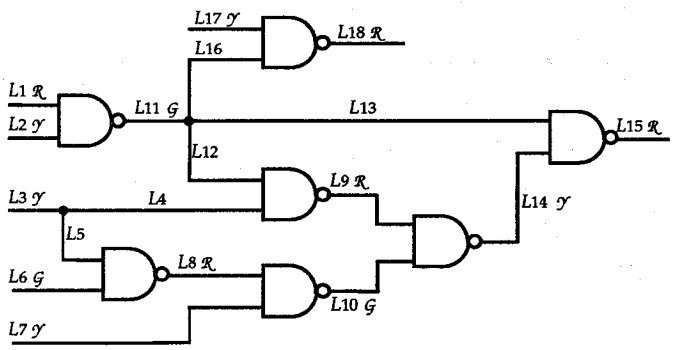


図 4.5　2-UC-NAND 回路の例

4.4.1　故障シミュレーションの方法

4.4.1.1　故障影響の伝搬

手続きTGで生成されたテスト集合を正常回路に印加したとき，各信号線に基本系列が現われる．その例を図4.6に示す．

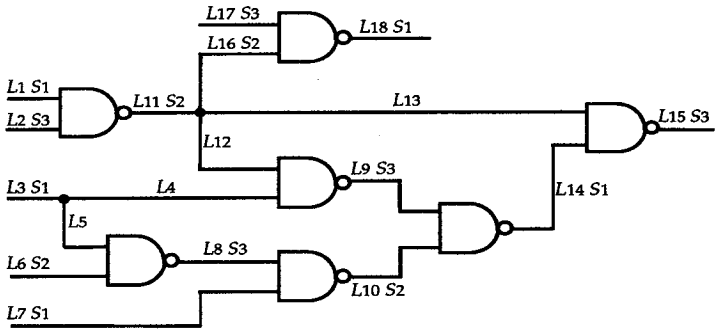


図4.6　2 - UC - NAND回路へのテスト系列の印加

あるゲートの入力線または出力線に縮退故障があれば，そのゲートの出力系列が基本系列でなくなる．そのゲートの出力線が他のゲートの入力線でもあるとき，その故障影響が回路内の他の信号線に伝わる可能性もある．その例を図4.7に示す．$L1$ に1縮退故障があるとき，$L1$ にある系列は $S1$ と異なる系列となり，$S1^*$ で表す．この故障の影響は $L11, L16, L18$ に伝わる．

故障影響は回路内の他の信号線に伝わることがあるが，必ずしも外部出力線まで伝わるとは限らない．すなわち，故障影響の伝搬が阻止される可能性はある．

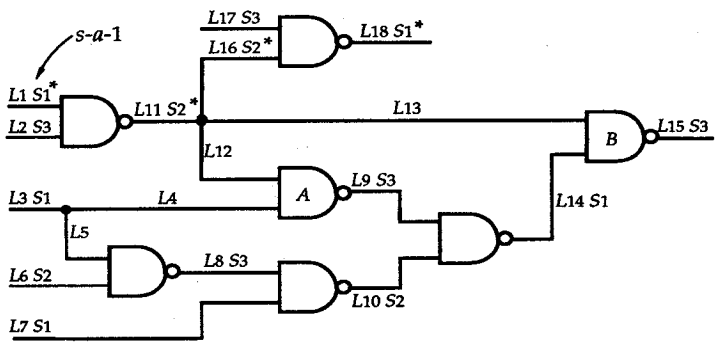


図4.7　故障2 - UC - NAND回路の例

定義4.9　回路に故障が存在するとき，入力線が異常系列をもち出力線が基本系列をもつようなゲートを，その故障の伝搬阻止ゲートと呼ぶ．

例えば，図4.7に示した回路において，ゲート A と B は伝搬阻止ゲートである．伝搬阻止ゲートの存在により，故障影響の伝搬は回路内で阻止される．故障が1縮退故障であるとき，あるゲートが伝搬阻止ゲートである必要十分条件は次の定理で与えられる．

定理4.4　2 - UC - NAND 回路において，1縮退故障をもつ信号線の色が C であるとする．あるゲートがその1縮退故障の伝搬阻止ゲートであるための必要十分条件は，そのゲートの1本の入力線の色が C であり，もう1本の入力線に異常系列があることである．

4.4.1.2　対象故障数の削減

ここでの故障シミュレーションの目的は，故障影響が外部出力線に伝わるような縮退故障を見い出すことである．シミュレーション時間を短縮するため，その影響が必ず外部出力線まで伝搬しないような故障を対象故障集合より取り除き，対象故障集合を小さくすることが望ましい．このような故障は伝搬阻止ゲートの必要十分条件を用いて簡単に発見できる．例えば，図4.8に示すように，ゲート G は2本の入力線 Li と Lj をもつ．Li は部分回路の出力線である．もし Lj が色 C をもつならば，部分回路内のすべての色 C をもつ1縮退故障が Li を通じて外部信号線まで伝搬しないことになる．これで，部分回路内のすべての色 C をもつ1縮退故障を対象故障集合から取り除くことができる．

NANDゲートのみからなる回路において，すべての1縮退故障と分岐の幹上にある0縮退故障を考えれば十分である．0縮退故障を処理するため，フロンティア・ゲートの概念を導入する．

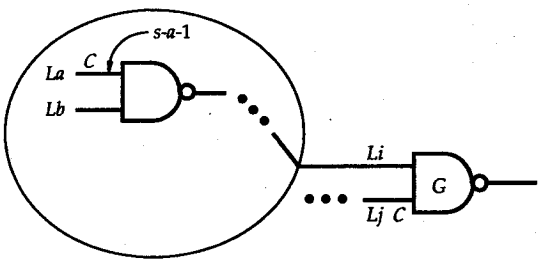


図4.8　伝搬阻止ゲートの効果

定義4.10　L を分岐の幹とする．あるゲートの入力線が L からの分岐であるとき，そのゲートを L のフロンティア・ゲートと呼ぶ．

明らかに，ある幹の0縮退故障はその幹のフロンティア・ゲートの出力線の1縮退故障

の等価故障である．したがって，その幹の0縮退故障の影響が外部出力線に伝わる必要条件は，その幹の少なくとも1個のフロンティア・ゲートの出力線の1縮退故障の影響が外部出力線に伝わることである．

以上に述べたことに基づき，対象故障数を削減する手続きを提案する．ここで，回路内の信号線に割当られた3種類の色を $C1, C2$ および $C3$ とする．また，P は伝搬（Propagation）を，NP は非伝搬（Non-Propagation）を表す．

手続き FNR（対象故障数を削減する）

(1) i を1とする．故障リスト FL を空にする．

(2) P を各外部出力線に割り当てる．

(3) 次の規則にしたがって，P または NP を残りの信号線に割り当てる．

規則 1： あるゲートの出力線が P をもち，かつ入力線にどの記号も割当られていないとする．入力線に色 Ci がなければ，P を入力線に割り当てる．1本の入力線に色 Ci があれば，P をその入力線に割り当て，NP をもう1本の入力線に割り当てる．あるゲートの出力線が NP をもち，かつ入力線にどの記号も割当られていないとする．この場合，NP を入力線に割り当てる．

規則 2： 分岐のすべての枝が NP をもち，かつその分岐の幹にどの記号も割当られていない場合，NP を分岐の幹に割り当てる．分岐の少なくとも1本の枝が P をもち，しかもその分岐の幹にどの記号も割当られていない場合，P をその幹に割り当てる．

(4) ゲート G の入力線を $L1$ と $L2$ とする．$L1$ が NP を，$L2$ が色 Ci と P を，そのゲートの出力線が P をもつならば，$L1$ の NP を P に変える．

(5) P をもつ信号線の1縮退故障を故障リスト FL に入れる．

(6) L を分岐の幹とする．その幹の少なくとも1個のフロンティア・ゲートの出力線が P をもつなら，L の0縮退故障を故障リスト FL に入れる．

(7) $i=i+1$ とする．$i \leq 3$ のとき，(1) に戻る．　□

例 4.3　図4.9に示す回路内の信号線の色 R に対して P と NP を割り当てる．明らかに，色 P に対して，$L5, L6, L7, L8$ の1縮退故障の影響は外部出力線に伝わらない．　□

手続き FNR を用いることにより，その影響が必ず外部出力線に伝わらない故障を発見することができ，これらの故障は対象故障集合から取り除かれる．

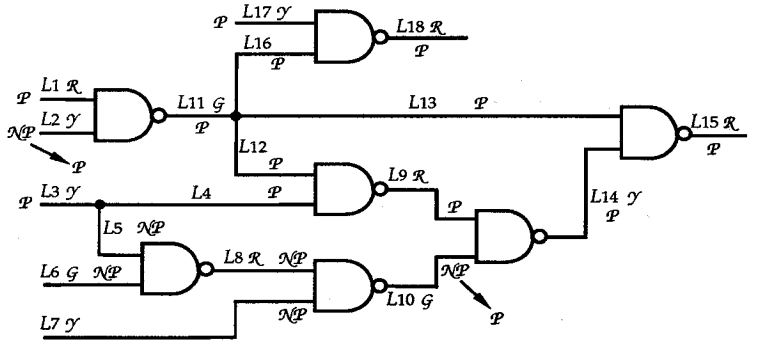


図 4.9　対象故障数の削減

4.4.1.3　故障シミュレーション手続き

以下では，ある信号線 L の 1 縮退故障 f の影響が外部出力線に伝わるか否かを調べる手続きについて述べる．f は手続き FNR により求められた対象故障リスト中にあるとする．このシミュレーションでは，基本系列を意味する N と異常系列を意味する E が用いられる．なお，信号線 L の色を C とする．

手続き FS（故障シミュレーションを行なう）

(1)　L から外部出力線まですべてのパスを求める．パスを L - $L1(L'1)$ - $L2(L'2)$ - . . . - $Lh(L'h)$ - Lpo で表す．ここで，Lpo は外部出力線である．Li が分岐の幹のとき，$L'i$ を空とする．Li が 2 入力 NAND ゲートの入力線のとき，$L'i$ がそのゲートへのもう 1 本の入力線とする（$i = 1, 2, \ldots, h$）．Li をパス線，$L'i$ をペア線と呼ぶ．

(2)　L に E を割り当てる．L から外部出力線までのすべてのパス上のパス線でないペア線に N を割り当てる．

(3)　N と E を次のように残りの信号線に割り当てる．パス線 Li に N または E が割り当てられており，パス線 Li-1 に N または E が割り当てられているとする．Li-1 にペア線がないとき，Li に Li-1 と同じシンボルを割り当てる．Li-1 にペア線があり，しかも Li-1 と $L'i$-1 が N をもつとき，Li に N を割り当てる．Li-1 または $L'i$-1 が E をもち，しかも Li-1 も $L'i$-1 も色 C をもたない場合，Li に E を割り当てる．Li-1（$L'i$-1）が E をもち，しかも $L'i$-1（Li-1）が色 C をもつ場合，Li に N を割り当てる．

(4)　少なくとも 1 本の外部出力線が E をもつなら，信号線 L の 1 縮退故障 f の影響は外部出力線に伝わる．　□

例 4.4　図 4.10 に示す回路の信号線 $L1$ の 1 縮退故障の影響が外部出力線に伝わるか否かを調べる．$L1$ から外部出力線までのパスは次に示す．

*L*1 - *L*3() - *L*4(*L*6) - *L*7() - *L*8(*L*10) - *L*11
*L*1 - *L*3() - *L*4(*L*6) - *L*7() - *L*9(*L*5) - *L*12
*L*1 - *L*3() - *L*5(*L*9) - *L*12

手続き *FS* の (2) を実行した結果を次に示す.

*L*1*E* - *L*3() - *L*4(*L*6*N*) - *L*7() - *L*8(*L*10*N*) - *L*11
*L*1*E* - *L*3() - *L*4(*L*6*N*) - *L*7() - *L*9(*L*5) - *L*12
*L*1*E* - *L*3() - *L*5(*L*9) - *L*12

手続き *FS* 内の (3) を実行した結果を次に示す.

*L*1*E* - *L*3*E*() - *L*4*E*(*L*6*N*) - *L*7*E*() - *L*8*E*(*L*10*N*) - *L*11*E*
*L*1*E* - *L*3*E*() - *L*4*E*(*L*6*N*) - *L*7*E*() - *L*9*E*(*L*5) - *L*12*N*
*L*1*E* - *L*3*E*() - *L*5*E*(*L*9) - *L*12*N*

これにより，信号線 *L*1 の 1 縮退故障の影響は外部出力線に伝わることが分かる. □

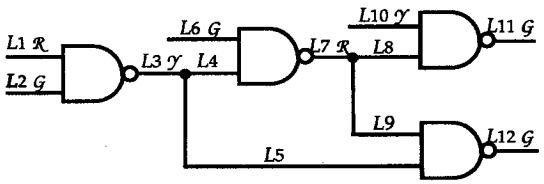


図 4.10　故障シミュレーションの例

分岐の幹の 0 縮退故障はその幹のフロンティア・ゲートの出力線の 1 縮退故障と等価である. その幹のフロンティア・ゲートが複数ある場合，少なくとも 1 個のフロンティア・ゲートの出力線の 1 縮退故障の影響が外部出力線に伝わるなら，分岐の幹の 0 縮退故障の影響も外部出力線に伝わる.

手続き FS を用いることにより，通常のテスト環境における，全可観測な環境で生成されたテスト集合の故障検出率を調べることができる.

4.4.2　テスト生成法

前節では，全可観測な環境で生成されたテスト集合を用いても通常のテスト環境で一部の故障を検出することができることを示した. そのテスト集合で検出できない故障について，

さらにテスト生成を行なう必要がある．このために，D, PODEM, FAN, CONT, SOCRATES などのテスト生成アルゴリズムを用いることができる．被検査回路は一般の回路ではなく，2 - UC - NAND 回路であるため，回路特徴を利用したアルゴリズムを用いることが考えられる．

2 - UC - NAND 回路において，一つの彩色解は一つのテスト集合に対応する．つまり，ある故障が検出されるように彩色解を求めればよい．このため，伝搬阻止ゲートの概念を用い故障影響の伝搬が阻止されないように彩色解を求める必要がある．この方法は従来のテスト生成アルゴリズム手法に似ている．

もう一つの方法では，2 - UC - NAND 回路のすべての彩色解を求めて，各彩色解に対応するテスト集合の和集合はその回路の通常のテスト環境でのテスト集合とする．

4.4.3　実験結果と考察

提案した故障シミュレーション手続きFSをC言語によりプログラム化し，実験を行なった．使用したコンピュータは富士通のS-4/LC（12.5 MIPS）である．実験用の回路はISCAS 1985 のベンチマーク組合せ回路である [71].

表4.2は全可観測な環境で生成されたテスト集合の，通常のテスト環境における故障検出率を示す．

表 4.2　実験結果

回路	故障検出率（%）	観測率（%）
2UC-17	94.1	33.3
2UC-ADDER	61.3	83.3
2UC-432	16.5	87.4
2UC-499	66.4	63.2
2UC-1908	19.8	87.7
2UC-2670	13.2	88.4
2UC-3540	9.8	93.4
2UC-5315	19.9	91.3
2UC7552	14.7	83.7

表4.2の結果は，全可観測な環境においても，100% の故障検出率を得るためにすべてのゲートの出力線が観測できるとしなくてもよいことを示している．必要な観測点数と全ゲート数の比率は表4.2に示されている．この情報は，クロスチエック法 [21, 22] などを用いて回路内に観測点を設けるときに有用である．

4.5　k - UCP 回路概念の拡張

全可観測な環境では，k-UCP 回路の故障検出と故障診断は短いテスト系列で行なうことができる．しかし，k-UCP 回路を構成するゲートとしては，k 入力 AND, OR, AND, NAND ゲートおよびNOTゲートしか使用できない．つまり，任意の組合せ回路を k-UCP 回路に

変換するとき，まず k 入力でないゲートを k 入力のゲートによる等価な部分回路に変換する必要がある．このため，ゲートの入力数に関する制限をなくすことが望ましい．以下では，k-R 回路を提案し，この問題を解決する．

4.5.1　k-R 回路の定義

定義 4.1 1　NOT ゲート，XOR ゲート，XNOR ゲート，および入力数が k 以下の OR, AND, NAND, NOR ゲートで構成される組合せ回路を拡張 k-U 回路と呼ぶ．

定義 4.1 2　拡張 k-U 回路において，表 4.3 に示すルールにしたがって，正極性（+）と負極性（−）をすべての信号線に割り当てることができるとき，この拡張 k-U 回路が正しい極性をもつという．

表 4.3　k-R 回路の極性割当ルール

	入力	出力
NAND	+	+
AND	+	-
NOR	-	-
OR	-	+
NOT	-/+	+/-
XOR	-/+	-/+
XNOR	-/+	-/+

定義 4.12 は定義 4.3 の拡張である．例えば，負極性を OR ゲートの入力線に，正極性を OR ゲートの出力線に割り当てなければならない．また，XOR ゲートまたは XNOR ゲートの入力線と出力線に同じ極性を割り当てる必要がある．

定義 4.1 3　$U=\{1, 2, \ldots, k+1\}$ とする．拡張 k-U 回路の各信号線に U の空でない部分集合を割り当てる．信号線 L に割当られた部分集合を L の重みと呼ぶ。回路内の各信号線の重みが次の条件を満たすとき、その回路が k-均衡性をもつという．

(1)　回路内の任意の t 入力の AND, OR, AND, NOR ゲートの入力線の重みを $W1, W2, \ldots, Wt$, 出力線の重みを W とすると，$W = U - (W1 + W2 + \ldots + Wt)$ と $Wi - (W1 + \ldots Wi\text{-}1 + Wi+1 + \ldots + Wt) \neq \emptyset$ が成り立つ．なお，$i = 1, 2, \ldots, t \leqq k$ とする．

(2)　回路内の任意の XOR または XNOR ゲートの入力線の重みを $W1, W2$ とすると，

$W1 - W2 \neq \emptyset$ と $W2 - W1 \neq \emptyset$ が成り立つ．

(3)　回路内の任意の NOT ゲートの入力線の重みと出力線の重みが等しい．

定義 4.1 4　正しい極性と k-均衡性をもつ拡張 k-U 回路を k-R 回路と呼ぶ．

例えば，図 4.11 に示す回路は 4-R 回路である．

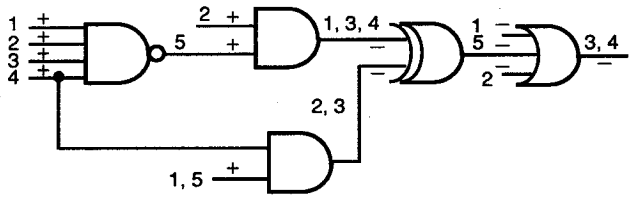


図 4.11　4-R 回路の例

4.5.2　k-基本系列の定義

k-UCP 回路と同様に，k-R 回路のテスト集合もあらかじめ生成されている基本系列から構成される．

定義 4.1 5　$U=\{1, 2, \ldots, k+1\}$, $W=\{w1, w2, \ldots, wn\} \subseteq U$ とする．長さ $k+1$ の論理 0 と 1 からなる系列 $e1, \ldots, ew1, \ldots, ew2, \ldots, ewn, \ldots, ek+1$ が $ei=1, ej=0$ $(j=w1, w2, \ldots, wn; i=1, 2, \ldots, k+1; i \neq j)$ という条件を満たすとき，その系列を集合 W の正の k-基本系列と呼び，$Pk(W)$ で表す．系列 NOT($Pk(W)$) を集合 W の負の k-基本系列といい，$Nk(W)$ で表す．

3-基本系列の例を次に示す．

$S(1)=0111$　　$T(1)=1000$

$S(1, 2)=0011$　　$T(1, 2)=1100$

$S(2, 3, 4)=1000$　　$T(2, 3, 4)=0111$

基本系列は次の性質をもつ．

補題 4.5　k-R 回路において，次の結論が成り立つ．

(1)　回路内の任意の t 入力 AND, NOR, NAND, NOR ゲートの入力線の重みを $W1, W2, \ldots, Wt$, 出力線の重みを W とすると，次の式が成り立つ．

$$\mathrm{AND}\ (Pk(W1), Pk(W2), \ldots, Pk(Wt)) = Nk(W)$$
$$\mathrm{OR}\ (Nk(W1), Nk(W2), \ldots, Nk(Wt)) = Pk(W)$$
$$\mathrm{NAND}\ (Pk(W1), Pk(W2), \ldots, Pk(Wt)) = Pk(W)$$
$$\mathrm{NOR}\ (Nk(W1), Nk(W2), \ldots, Nk(Wt)) = Pk(W)$$

(2)　回路内の任意の XOR または XNOR ゲートの入力線の重みを $W1$, $W2$, 出力線の重みを W とすると，次の式が成り立つ．

$$\mathrm{XOR}\ (Pk(W1), Pk(W2)) = Pk(W)$$
$$\mathrm{XOR}\ (Nk(W1), Nk(W2)) = Nk(W)$$
$$\mathrm{XNOR}\ (Pk(W1), Pk(W2)) = Nk(W)$$
$$\mathrm{XNOR}\ (Nk(W1), Nk(W2)) = Pk(W)$$

(3)　回路内の任意の NOT ゲートの入力線または出力線の重みを W とすると，次の式が成り立つ．

$$\mathrm{NOT}\ (Pk(W)) = Nk(W), \quad \mathrm{NOT}\ (Nk(W)) = Pk(\mathrm{W})$$

（証明）　定義 4.13 より，結論 (3) が明らかである．次に，NAND $(Pk(W1), \mathrm{Pk}(W2), \ldots, Pk(Wt)) = Pk(W)$ が成立することを証明する．他の式については，ほぼ同様に証明できる．

$Pk(Wi), Pk(W2), \ldots, Pk(Wt)$ は論理値 0 と 1 からなる系列である．$Pk(W1)$ などの i ビット目を $Pk(\mathrm{W1}, i)$ などで表す．一般性を失うことなく，$Pk(W1, i) = 0$ と仮定する．$W = U - W = U - (W1 + \ldots Wi\text{-}1 + Wi\text{+}1 + \ldots + Wt) \neq \emptyset$ と $i \in W$ であるため，$i \in W1$ である．つまり，$Pk(W1, i) = 0$ である．NAND $(Pk(W1, i), Pk(W2, i), \ldots, \mathrm{Pk}(Wt, i)) = 1$ であるため，NAND $(Pk(W1, i), Pk(W2, i), \ldots, Pk(Wt, i)) = Pk(W, i)$ である．なお，$i = 1, 2, \ldots, k + 1$ とする．よって，NAND $(Pk(W1), Pk(W2), \ldots, Pk(Wt)) = Pk(W)$ である．　□

補題 4.6　k-R 回路において，次の結論が成り立つ．

(1)　回路内の任意の t 入力 AND, NAND（OR, NOR）ゲートの入力線の重みを $W1, W2, \ldots, Wt$ とすると，$Pk(W1), Pk(W2), \ldots, Pk(Wt)$（$Nk(W1), Nk(W2), \ldots, Nk(Wt)$）を印加することにより，そのゲートのすべての縮退故障を検出することができる．

(2)　回路内の任意の XOR または XNOR ゲートの入力線の重みを $W1$, $W2$ とすると，$Pk(W1)$ と $Pk(W2)$ または $Nk(W1)$ と $Nk(W2)$ を印加することにより，そのゲートのすべての縮退故障を検出することができる．

(3)　回路内の任意の NOT ゲートの入力線または出力線の重みを W とすると，$Pk(W)$ または $Nk(W)$ を印加することにより，そのゲートのすべての縮退故障を検出することができる．

（証明）　図 4.12 に示すように，t 入力 NAND ゲートの入力線の重みを $W1, W2, \ldots, Wt$ とすると，$Pk(W1), Pk(W2), \ldots, Pk(Wt)$ を印加することにより，そのゲートのすべての縮退故障を検出することができることを証明する．他の結論はほぼ同様に証明できる．

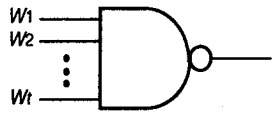


図 4.12　t 入力 NAND ゲート

$W = U - (W1 + \ldots Wt\text{-}1 + Wi\text{+}1 + \ldots + Wt) \neq \emptyset$ であるため，次の式を満たすような $w1, w2, \ldots, wt, w$ が必ず存在する．

$$w1 \in W1, w1 \notin W2, \ldots, w1 \notin Wt$$
$$w2 \notin W1, w2 \in W2, \ldots, w2 \notin Wt$$
$$\ldots$$
$$wt \notin W1, wt \notin W2, \ldots, wt \in Wt$$
$$w \notin W1, w \notin W2, \ldots, w \notin Wt$$

これで，図 4.13 に示すように，t 入力 NAND ゲートのすべての縮退故障を検出するようなベクトルが $Pk(W1), Pk(W2), \ldots, Pk(Wt)$ に含まれることが分かる．図 4.13 では，$w1, w2, \ldots, wt, w$ がビットの位置を意味する．つまり，$S(W1), S(W2), \ldots, S(Wt)$ にはそのゲートのすべての縮退故障を検出するようなベクトルが含まれている．よって，t 入力 NAND ゲートのすべての縮退故障は $Pk(W1), Pk(W2), \ldots, Pk(Wt)$ を印加することにより検出できる．　□

	$w1$	$w2$		wt	w
$S(W1)$	0	1	•••	1	1
$S(W2)$	1	0	•••	1	1
⋮	⋮	⋮	•••	⋮	⋮
$S(Wt)$	1	1	•••	0	1

図 4.13　補題 4.6 の証明

4.5.3　縮退故障の検出

補題 4.6 より，k-R 回路において，重み W と正極性（負極性）をもつ外部入力線に基本系列 $S(W)$（$T(W)$）を印加することにより，すべての縮退故障が検出できる．基本系列の長さが $k+1$ であるので，次の定理が成立する．

定理 4.5　全可観測な環境において，k-R 回路のすべての縮退故障は長さ $k+1$ のテスト系列で検出できる．

図 4.14 に 3 - R 回路の例とそのテスト集合を示す.

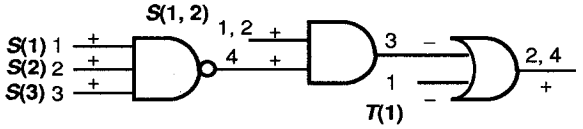


図 4.14　テスト集合の例

4.6　まとめ

本章では，全可観測な環境でのテスト容易な組合せ回路として，k - UCP 回路および k - R 回路を提案した.

k - UCP 回路は NOT ゲートと k 入力のゲートで構成され，いくつかの構造上の制約も受ける．そのため，k - UCP 回のすべての縮退故障とスタック・オープン故障は，全可観測な環境においてそれぞれ長さ $k+1$ と $k(k+1)+1$ のテスト系列で検出できる．実際の k の値が 2 または 3 なので，テスト系列は非常に短い.

k - UCP 回路のテスト集合はあらかじめ生成されている基本系列より構成される．基本系列の長さが回路のテスト系列の長さなので，なるべく短い基本系列を生成しなければならない．本章では，縮退故障用とスタック・オープン故障用の短い基本系列の生成手法についても述べた.

次に，k - UCP 回路のすべての縮退故障を検出するには，一部のゲートの出力線を観測するだけでよいことを明らかにした．それは，k - UCP 回路にテスト集合を印加したとき，一部の縮退故障の影響は外部出力線に伝わるからである．実験結果では，最多 93.4%，最少 33.3% のゲートの出力線を観測すれば十分である．このような情報は，クロスチエック法などで回路内に観測点を設けるとき有用である.

さらに，k - UCP 回路におけるゲートの入力線数が k であるという制約をなくすため，k - R 回路を提案した．k - R 回路は NOT ゲート，および入力線数が k 以下の基本ゲートで構成される．k - R 回路のすべての縮退故障は，全可観測な環境において長さ $k+1$ のテスト系列で検出できる．k - R 回路は一般的な回路で，k - UCP 回路，k - UC - NAND 回路および k - UC - NOR 回路などをそれの特例と見なすことができる.

5 全可観測な環境での組合せ回路のテスト容易化設計

5.1 概説

任意の組合せ回路は必ずしも k-UCP 回路ではない．例えば，図 5.1 の 2-U 回路は 3 色解をもたない．なぜなら，外部入力線にどの色を塗っても $L16$ と $L17$ が同じ色になるからである．また，この回路は 2-UP 回路でもない．なぜなら，ゲート G の入力線に正極性が必要なのにゲート F が負極性を提供しているからである．OR ゲート H をゲート G と F の間に挿入することにより，図 5.1 の回路は図 4.1 に示した 2-UCP 回路に変換できる．

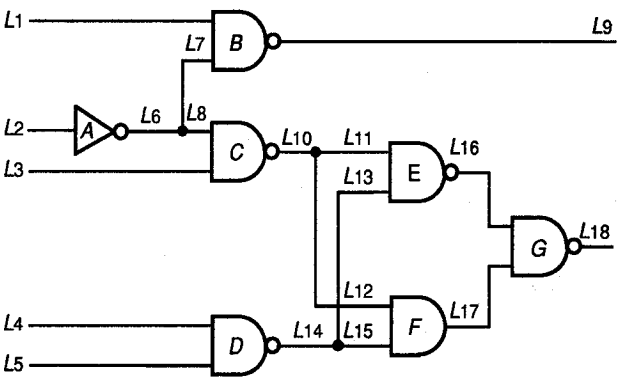


図 5.1　3 色解と正しい極性をもたない回路

このように，任意の組合せ回路は k-UCP 回路に変換できるが，オーバーヘッドの問題および k-U 回路が $k+1$ 色解をもつか否かを決定する問題がある．本章では，これらの問題を考慮して回路変換法を提案する [38, 40].

回路変換の目標は，なるべく少ないオーバーヘッドで与えられた組合せ回路を k-UCP 回路に変換することである．一般に，回路変換のオーバーヘッドはゲート数で評価されるが，これは CMOS 回路の場合に必ずしも適当ではない．例えば，4 入力 NAND ゲートは図 5.2 に示すように 2 入力のゲートで構成されるように変換できる．ゲート数は 1 から 3 に増え

るが，トランジスタ数は8から16に増える．増加率はゲート数では200%であるが，トランジスタ数では100%である．一般に，トランジスタ数はチップ面積をゲート数より正確に反映するので，本論文では，トランジスタ数で回路変換のオーバーヘッドを評価する．

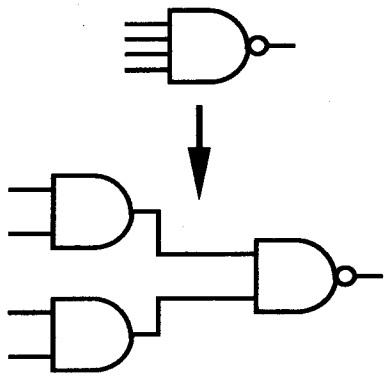


図5.2　4入力NANDゲートの分解

k-UCP回路への変換は3段階に分けて行なわれる．

(1)　与えられた組合せ回路からk-U回路への変換
(2)　k-U回路からk-UP回路への変換
(3)　k-U回路からk-UC回路への変換

明らかに，これらの変換を(1)→(2)→(3)の順に実行した結果はk-UCP回路であり，(1)→(3)→(2)の順に実行した結果もk-UCP回路である．

5.2　k-U回路への変換

この変換では，与えられた組合せ回路をk-U回路に変換する．ここではまず，kの値を決める必要がある．kの値を決める方法は2通りある．第1は設計者が指定する方法であり，第2はk-U回路に変換するための付加トランジスタ数が最小になるように決める方法である．2番目の方法ではまず，与えられた回路のすべてのゲートのファンイン数を調べ，その最大値をf_{MAX}とする．合理的なkの値は2とf_{MAX}の間にあるので，各kに対してファンイン数を調整するのに必要な付加トランジスタの数を計算し，付加トランジスタの数が一番少ないkの値を最終のkの値とする．

kの値が決まれば，次はNOTゲート以外の各ゲートのファンイン数がすべてkになるように変換を行なう．例えば，4入力NANDゲートについては，図5.2に示したように2入力のゲートによる構成におきかえる．さらに，XOR, XNORゲートなどのAND, OR, NAND, NORおよびNOTゲート以外のゲートをAND, OR, NAND, NORおよびNOTゲートで構成

されるように変換する．例えば，XORゲートは，図5.3に示すように2入力NANDゲートからなる部分回路におきかえることができる．

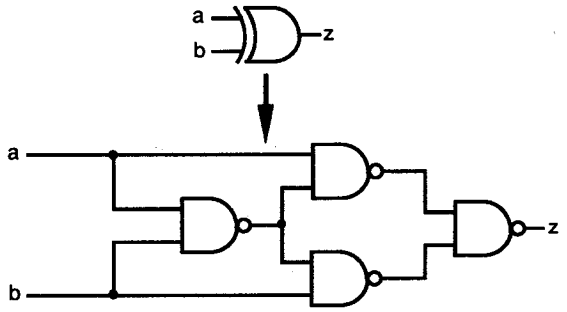


図5.3　XORゲートの2入力NANDゲートによる表現

k-U回路への変換は難しい問題ではないが，変換のオーバヘッドを少なくするための工夫が必要である[48]．明らかに，nを回路のゲート数とすると，この変換の計算量はO(n)である．

5.3　k-UP回路への変換

この変換では，極性の調整を行なうことにより，k-U回路をk-UP回路に変換する．具体的にはまず，k-U回路の各信号線に表4.1に示したルールにしたがって極性を割り当てる．同じ信号線に異なる極性が割り当てられたとき，図5.4に示すようにその信号線にANDまたはORゲートを挿入して極性の調整を行なう．nを回路のゲート数とすると，この変換の計算量はO(n)である．

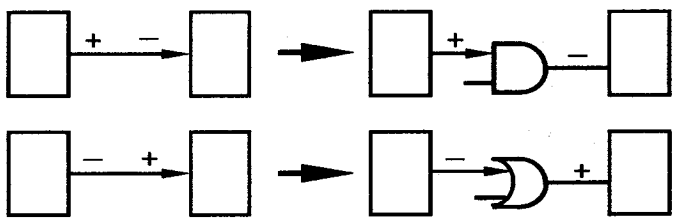


図5.4　極性の調整

5.4　k-UC回路への変換

この変換を行なうとき，まず与えられたk-U回路が$k+1$彩色解をもつか否かを決定する必要がある．

色 $C1, C2, \ldots, Ck$ が k 入力のゲートの入力線に割り当てられたとき，そのゲートの出力線に割り当てるべき色は明らかに $Ck+1$ である．このように，これまでの外部入力線への色の割り当てにより色が一意的に決まる信号線を探し出す操作を色の含意操作と呼ぶ．回路のすべての外部入力線に色が割り当てられたとする．色の含意操作により他のすべての信号線にも色が割り当てられるようになれば，この回路は $k+1$ 彩色解をもつことが分かる．すべての外部入力線に対する任意の色の割り当てのもとで色の含意操作を行なっても，少なくとも1本の信号線には色を割り当てることができなければ，この回路は $k+1$ 彩色解をもたないことが分かる．

このことより，k-U 回路が $k+1$ 彩色解をもつか否かを決定する問題は，n 次元空間での探索問題として形式化できることが分かる．k-U 回路が $k+1$ 彩色解をもつか否かを決定する問題を単に $k+1$ 彩色解問題と呼ぶ．以下では，$k+1$ 彩色解問題が NP 完全であることを証明する．$k+1$ 彩色解問題が NP 完全であることは，それを解く多項式時間のアルゴリズムを発見すれば，すべての NP 完全問題を解く多項式時間のアルゴリズムを発見したことを意味する．つまり，$k+1$ 彩色解問題を解く多項式時間のアルゴリズムを発見することは極めて困難である．現実的には，NP 完全問題をヒュリスティックな方法で解決せざるを得ない．

5.4.1　$k+1$ 彩色解問題

本節では，$k+1$ 彩色解問題が NP 完全であることを証明する．一般に，ある問題が NP 完全であることの証明は2段階に分けて行なわれる [49, 50]．まず，その問題が NP 問題である，つまり多項式時間の非決定性アルゴリズムで解けることを証明する．次に，ある既知の NP 完全問題が多項式時間のアルゴリズムでその問題に帰着できることを示せばよい．

補題 5.1　$k+1$ 彩色解問題は NP 問題である．

（証明）　与えられた k-U 回路の外部入力線へのすべての色の割り当ての集合を S とする．$k+1$ 彩色解問題を次の非決定性アルゴリズム AL1 により解くことができる．

アルゴリズム AL1　（$k+1$ 彩色解問題を解く）

(1)　集合 S から非決定的に一つの色の割り当て s を選ぶ．

(2)　s のもとで色の含意操作によりすべての信号線に矛盾なく色が割り当てられれば，その回路が $k+1$ 彩色解をもつ．そうでなければ，その回路が $k+1$ 彩色解をもたない．　□

アルゴリズム AL1 が明らかに多項式時間のアルゴリズムなので，彩色解問題は NP 問題であることが分かる．

定義 5.1　グラフ$G=(V,E)$と整数の集合$\{1,2,\ldots,m\}$が与えられたとする．$(u,v)\subset E$に対して$f(u)\neq f(v)$となるような関数$f: V\rightarrow\{1,2,\ldots,m\}$が存在するとき，グラフ$G$は$m$彩色可能であるという．任意のグラフ$G$と$\{1,2,\ldots,m\}$に対して，$G$が$m$彩色可能か否かを決定する問題を$m$彩色可能性問題と呼ぶ[49, 50].

グラフのm彩色可能性問題はNP完全であることがすでに証明されている[49, 50]．以下では，グラフの$k+1$彩色可能性問題が多項式時間のアルゴリズムで$k+1$彩色解問題に帰着できることを証明することにより，$k+1$彩色解問題もNP完全であることを示す．そのため，まず，次のアルゴリズムでk-U回路をグラフに変換する．

アルゴリズム AL2（k-U回路をグラフに変換する）

(1)　回路内のゲートの出力線を頂点とする．

(2)　ゲートの入力線を表す頂点を辺で結ぶ．

(3)　ゲートの出力線を表す頂点とそのゲートの入力線を表す頂点を辺で結ぶ．　□

AL2の実行例を図5.5に示す．AL2で得られるグラフを回路グラフと呼ぶ．

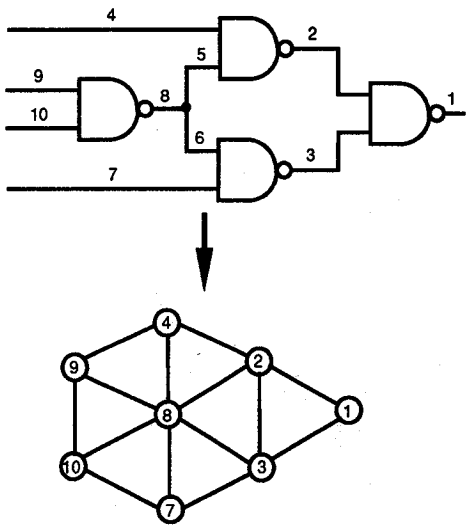


図5.5　回路グラフの例

補題 5.2　グラフの$k+1$彩色可能性問題が多項式時間のアルゴリズムで$k+1$彩色解問題に帰着できる．

（証明）　ここで，グラフの3彩色可能性問題は2-U回路が3彩色解をもつか否かを決定する問題に帰着できることを証明する．$k\geq 3$の場合についてほぼ同様に証明できる．

明らかに，アルゴリズムAL2は多項式時間で実行できる．また，回路グラフから2-U

回路を求める多項式時間のアルゴリズムも存在する．さらに，アルゴリズムAL2から分かるように，回路グラフが3彩色可能である必要十分条件はその2-U回路が3彩色解をもつことである．よって，グラフの3彩色可能性問題が多項式時間のアルゴリズムで3彩色解問題に帰着できる．□

補題5.1と補題5.2から明らかなように，次の定理が成り立つ．

定理5.1　$k+1$彩色解問題はNP完全である．

k-UC回路への変換ではまず，k-U回路が$k+1$色解をもつか否かを決定する必要がある．しかし，この$k+1$彩色解問題はNP完全問題であるため，多項式時間で解決するアルゴリズムを発見するのは極めて困難である．つまり，ヒューリスティックな手法を用いてこの問題の解決を試みざるを得ない．以下ではまず，$k+1$彩色解問題の規模を小さくする手法について述べる．次に，$k+1$彩色解問題のヒューリスティックな手法を用いた解決法を提案する．

5.4.2　回路の対象部分の抽出

$k+1$彩色解問題を考える場合，回路全体を対象とする必要はない．樹枝状の回路は明らかに彩色解をもつので，図5.6に示すAとBのような樹枝状の部分回路を取り除いてから$k+1$彩色解問題を考えれば十分である．

定義5.2　入力線がすべて外部入力線である樹枝状の部分回路を取り除いたあとの部分回路をもとの回路の対象部分と呼ぶ．

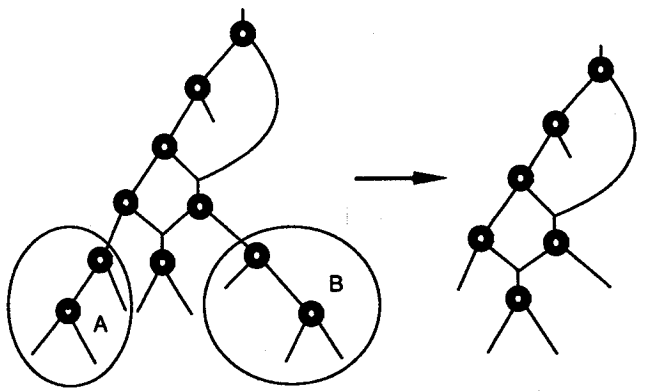


図5.6　回路の対象部分の例

このように，回路の対象部分を抽出することにより，回路の規模が小さくなり，問題解決の所要時間も少なくなる．

5.4.3　回路変換手法

彩色解問題は n 次元空間での探索問題として形式化できる．そこで，この問題を解く基本的な考え方は，外部入力線へのすべての色の割り当てを試してみて，彩色解をもたらすような色の割り当てが存在するかどうかを調べることである．k-U 回路において，1本の外部入力線に対して $k+1$ 個の可能な色の割り当てがある．すべての信号線に対する色の割り当ての数は，$(k+1)^n$ である．つまり，この探索操作を工夫しなければ，回路規模が少しでも大きくなると時間がかかり過ぎて実用的でなくなる．

提案する手法の基本的な考え方は次の通りである．一度に1本の外部入力線を選び，それにある色を割り当て，色の含意操作を行なう．色の矛盾があれば，その外部入力線に残りの可能な色の中から一つ選んで割り当てる．色の矛盾とは，同じゲートの少なくとも2本の入力線に同じ色が割り当てられていることである．もしその外部入力線のすべての可能な色が試されたら，その直前に色が割り当てられた外部入力線に残りの可能な色の中から一つ選んで割り当てる．もし最初に色を割り当てられた外部入力線のすべての可能な色が試されたら，その回路は $k+1$ 彩色解をもたないことが分かる．もし矛盾がなければ，次の外部入力線を選んで，同様の操作をしていく．

この過程において，色の含意操作は大きな役割をもつ．まず，それにより速く色の矛盾を発見することができる．次に，色の含意操作により外部入力線の色が自動的に決まるかまたは割り当て可能な色が限定されることになる．これより，外部入力線の色は速く決まることになる．

色の含意操作には不確定含意操作と確定含意操作の2種類がある．例えば，図5.7の2入力 NAND ゲートの入力線 a に色1を割り当てると，入力線 b と出力線 c に割り当て可能な色は色2と3のみとなる．これを図5.7では(2, 3)で示す．つまり，信号線 a に色を割り当てたことにより，他の信号線（b と c）の色が決まらなくても，割り当て可能な色の数が少なくなる．これは不確定含意操作の例である．図5.8に確定含意操作の例を示す．信号線 $L1$ と $L3$ にそれぞれ色 $C1$ と $C2$ を割り当てたとする．不確定含意操作で，信号線 $L4$ の可能な色は $(C2, C3)$ と $(C1, C3)$ である．したがって，$L4$ の色は必ず $C3$ である．これにより，$L2$ の色は必ず $C2$ であり，$L5$ の色は必ず $C1$ である．これは確定含意操作の例である．

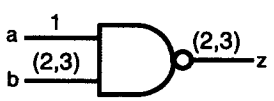


図5.7　不確定含意操作の例

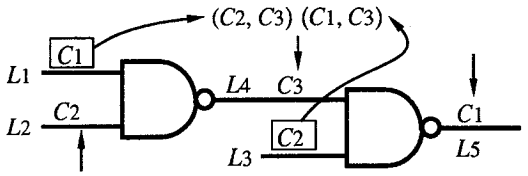


図 5.8　確定含意操作の例

$k+1$ 彩色解問題を解決する方法では，色の矛盾を発見したら，バックトラックを起こして探索を続ける．実行時間を短縮するため，バックトラックの回数を少なくする必要がある．このため，外部入力線を選択していく順番が重要である．以下では，回路の信号線にレベルを付け，それを利用して外部入力線の選択を行なう．

手続き LA（各信号線にレベルを割り当てる）

(1)　外部出力線にレベル 1 を付ける．

(2)　NOT ゲートの入出力線のレベルを等しくする．

(3)　k 入力ゲートの入力線のレベルをそれの出力線のレベルより一つ大きくする．

(4)　分岐の幹のレベルはその枝のレベルの中で一番大きいものと等しくする．　□

外部入力線のヒューリスティックな選択方法を次に示す．

手続き ES（外部入力線を選択する）

(1)　まだ色が決っていない外部入力線の中から，一番大きいレベルをもつ信号線を選択し，それに色を割り当てる．

(2)　一番大きいレベルをもつ外部入力線が2本以上ある場合，残りの割り当て可能な色の種類の少ない信号線を選択し，それに色を割り当てる．　□

次のアルゴリズムでは，k-U 回路が $k+1$ 彩色解をもつか否かを調べ，彩色解をもたない場合，図 5.9 に示すようにゲートを挿入して回路変換を行なう．

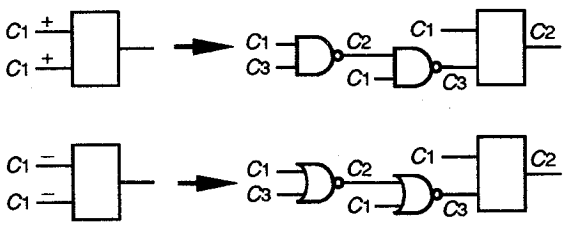


図 5.9　色の調整

手続き CM（彩色解の決定と回路変換を行なう）

(1)　まだ色が決まっていない外部出力線から手続き ES により1本の信号線を選んで一つの色を割り当て，この信号線と色をスタック L に入れる．残りの割り当て可能な色とこの信号線をスタック S に入れ，(2) へ移る．すべての外部出力線に色が割り当てられたら，回路が彩色解をもつとして終了する．

(2)　色の含意操作を行なう．色の矛盾がなければ (1) に戻る．色の矛盾があれば，(3) へ移る．

(3)　今回の色の割り当てを無効にする．S のトップにある要素を取り出し，それが示す信号線にそれが示す色を割り当て，(2) に戻る．S が空であれば，彩色解が存在しないことが分かり，(4) へ移る．

(4)　L のトップにある要素が示す信号線と色で信号線に色を割り当て，色の含意操作を行なう．色の矛盾については，ゲートを挿入することによりそれを解消する．(1) に戻る．　□

5.5　実験結果と考察

提案した回路変換手法を C 言語によりプログラム化し，回路変換の実験を行なった．使用したコンピュータは富士通の S-4/LC（12.5 MIPS）である．実験用の回路は ISCAS 1985 のベンチマーク組合せ回路である [71].

実験結果は表 5.1 に示す．実験 1 と 2 において，k の値は 2 である．これはベンチマーク回路を構成するゲートの大部分が 2 入力のゲートであるからである．実験 1 では，まずもとの回路を 2 - U 回路に変換し，さらに 2 - UP 回路に変換し，最後に 2 - UC 回路に変換した．その結果は AND, OR, NAND, NOR, NOT などの各種のゲートを含む k - UCP 回路である．実験 2 では，まずもとの回路を 2 入力 NAND ゲートからなる回路に変換し，さらに 2 - UC 回路に変換した．その結果は 2 - UC - NAND 回路である．表 5.1 の "2 - UC" は，実験 2 での 2 入力 NAND ゲートで構成される回路から 2 - UC 回路への変換のオーバーヘッドを示している．

表 5.1　実験結果

回路	トランジスタ数	オーバーヘッド (%)		
		実験 1	実験 2	2-UC
mc432	896	154.9	84.4	7.3
mc499	2180	41.1	26.2	4.6
mc880	1802	116.0	56.0	6.3
mc1355	2308	32.6	17.2	2.8
mc1908	3430	85.4	39.0	3.8
mc2670	5364	84.5	33.4	3.6
mc3540	7504	92.2	54.9	3.4
mc5315	11262	85.9	52.2	4.2
mc7552	15396	69.0	36.7	7.5

実験結果から明らかなように，実験2のオーバーヘッドは実験1のそれより小さい．また，表5.1の"2-UC"のオーバーヘッドは非常に小さい．つまり，与えられた回路はk入力NANDゲートのみを含むとき，それをk-UCP回路に変換するためのオーバーヘッドが非常に小さい．

5.6　まとめ

任意の組合せ回路をk-UCP回路に変換する手法について述べた．この変換は3段階に分けて行なわれる．まず最初の段階で，与えられた回路をk-U回路に変換する．次の2段階で，k-U回路をk-UP回路とk-UC回路に変換する．この中でもっとも困難な変換はk-U回路からk-UC回路への変換である．この変換ではまず，k-U回路が$k+1$色解をもつか否かについて決定する必要がある．この決定問題はNP完全問題であることを証明した．本章では，ヒューリスティックな手法でこの問題の解決を図った．基本的な考え方は，外部入力線へのすべての色の割り当ての中から彩色解をもたらすようなものを探索することである．この探索を速くするために，外部入力線のヒューリスティックな選び方，および色の含意操作などを提案した．

実験結果より次のことが分かった．

(1)　与えられた回路をk-UC-NANDまたはk-UC-NOR回路に変換するためのオーバーヘッドは様々なゲートからなる一般のk-UCP回路に変換するためのオーバーヘッドより少ない．

(2)　与えられた回路がk入力NANDまたはNORゲートで構成される場合，それをk-UC-NAND回路またはk-UC-NOR回路に変換するためのオーバーヘッドが非常に小さく，実用範囲内である．実際の論理回路の設計では，ゲートアレイ形式の設計をはじめ，2入力NANDまたはNORゲートが多用される場合が多いため，k-UCP回路の概念は論理回路の実際のテストに役立つと思われる．

6 全可観測な環境での順序回路のテスト

6.1 概説

実際に使用される論理回路の大部分は組合せ回路ではなく，順序回路である．2.4.1 で述べたように，順序回路のテストは組合せ回路のそれよりはるかに困難である．

第 4 章では，全可観測な環境でのテスト容易な組合せ回路について述べた．与えられた組合せ回路は k - UCP 回路であれば，短いテスト系列で故障検出できることを明らかにした．本章では，第 4 章で得られた結果をもとに，全可観測な環境でのテスト容易な順序回路について述べる [42, 43].

ここでは，クロック線を除く各信号線の縮退故障を対象故障とする．また，D - フリップ・フロップ を記憶素子とする．以下では，D - フリップ・フロップ のことを単にフリップ・フロップ と呼ぶ．一般の同期式順序回路を図 6.1 に示す．

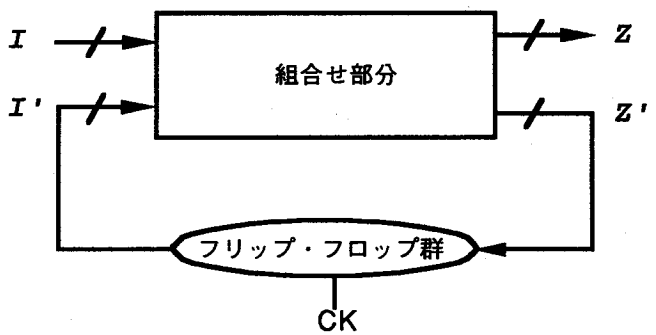


図 6.1　一般の順序回路

図6.1 より明らかなように，フリップ・フロップの入出力線は組合せ部分の出入力信号線でもある．つまり，組合せ部分に注目し，その入力線より組合せ部分に所要のテストベクトルを印加すれば，回路全体の縮退故障が検出されることになる．また，組合せ部分を短いテスト系列でテストするため，それを k - UCP 回路に変換することが考えられる．

しかし，この方法には大きな問題点がある．それは，フリップ・フロップの出力線を通じて順序回路の k - UCP 組合せ部分に所要のテストベクトルを印加することは必ずしも簡単であるとは限らない．このような印加が簡単にできるように，さらに回路構造を制限する必要がある．本章では，このような印加が簡単にできる順序回路として，k-UCP順序回路と k - UCP スキャン回路を提案する．全可観測な環境では，いずれの回路においても，少ないテストベクトルで故障検出と故障診断を行なうことができる．また，所要のテストベクトルは連続的に繰り返し印加することができる．

本章ではまず，順序回路の組合せ部分を k - UCP 回路に変換してテストを行なうときの問題点を明らかにする．次に，k - UCP 順序回路と k - UCP スキャン回路の概念およびテスト方法について述べる．以下では，縮退故障の基本系列のことを単に基本系列と呼ぶ．

6.2　問題点

すでに述べたように，順序回路の組合せ部分への入力線を通じて組合せ部分に所要のテストベクトルを印加すれば，回路全体の縮退故障がテストされることになる．組合せ部分への入力線は外部入力線の他にフリップ・フロップの出力線である場合もあるため，順序回路の k-UCP組合せ部分に所要のテストベクトルを印加することは必ずしも簡単であるとは限らない．これを次の例で具体的に説明する．

例 6.1　図 6.2 の順序回路の組合せ部分は 2 - UCP 回路に変換されている．この回路の組合せ部分をテストするために，$F1$ の出力線を通じて基本系列 $S2$ を，$F2$ の出力線を通じて基本系列 $S1$ を印加する必要がある．図 6.2 に示したように，$L1$ と $L4$ を通じて $S1$ と $S2$ の 1 ビット目が印加されたとする．このとき，$F1$ の入力線の値が 1 となる．つまり，次のクロック・パルスが印加されると，$F1$ の出力線の値が 1 となる．したがって，$F1$ の出力線を通じて基本系列 $S2 = 101$ を印加することが不可能である．　□

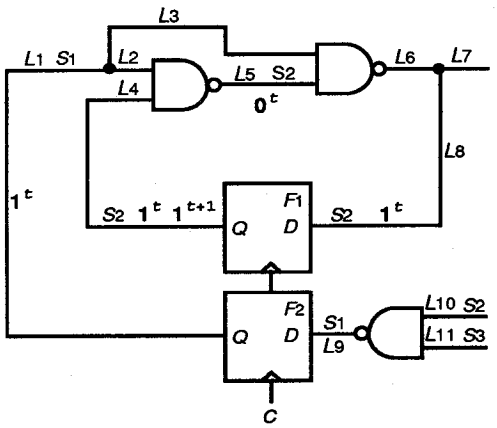


図 6.2　基本系列が印加不能な例

順序回路がスキャンパスをもつ場合，そのスキャンパスを通じて組合せ部分に所要のテストベクトルを印加することができるが，必要でないベクトルも印加しなければならない可能性がある．スキャンパスの長さを n とすれば，$k+1$ 個のテストベクトルを組合せ部分に印加するため，最悪の場合，$k(n-1)$ 個の必要でないベクトルが印加されてしまう．n が大きければ，テストベクトルの印加時間が長くなってしまう．また，電子ビームテスタを使用するときに必要な短い周期の周期動作を確立できない問題点もある．

つまり，順序回路の場合における主な問題点は，フリップ・フロップの出力線を通じて，所要の基本系列を組合せ部分に印加することが困難かまたは不可能なことである．以下では，この問題を解決するためのテスト容易な順序回路を提案する．

6.3　k-UCP 順序回路

6.3.1　基本概念

k-UCP 順序回路の基本的な考え方は，順序回路の組合せ部分を k-UCP 回路に変換し，さらにフリップ・フロップ周りの構造を限定することにより，k-UCP 組合せ部分へのテストベクトルの印加を簡単にすることである．

定義 6.1　以下の条件を満たす順序回路を k-UCP 順序回路と呼ぶ．

(1)　組合せ部分が k-UCP 回路である

(2)　各フリップ・フロップの入力線は以下のいずれかの種類に属する．

(2-a)　フリップ・フロップの入力線はある部分回路の出力線である．その部分回路は k-UCP 組合せ回路でかつすべての入力線が外部入力線である．このようなフリップ・フロップを I 型フリップ・フロップと呼ぶ．

(2-b)　フリップ・フロップの入力線はある部分回路の出力線である．その部分回路は図 6.3 に示すように直列につながる 2 個の k 入力 NAND（NOR）ゲートで構成され，$2(k-1)$ 本の外部入力線をもつ．それらの外部入力線を図 6.3 に示すようにそれぞれ制御線，伝搬線，および印加線と呼ぶ．このようなフリップ・フロップを NAND (NOR) II 型フリップ・フロップと呼ぶ．その 2 個のゲートを入力ゲートと呼ぶ．

例えば，図 6.4 に示す回路は明らかに 2-UCP 順序回路である．

明らかに，k-UCP 順序回路の組合せ部分への入力線は 3 種類に分けられる．つまり，I 型フリップ・フロップの出力線，II 型フリップ・フロップの出力線，および外部入力線である．

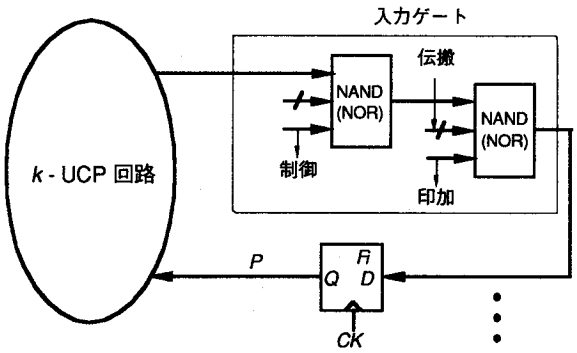


図 6.3　k - UCP 順序回路の定義

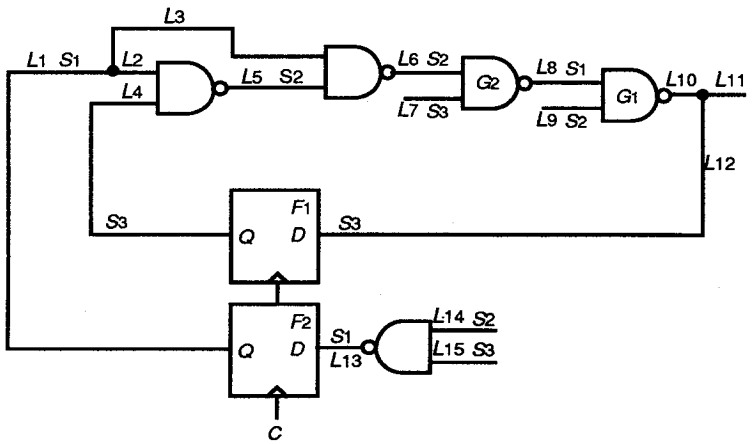


図 6.4　2 - UCP 順序回路

6.3.2　テスト方法

全可観測な環境において，k - UCP 順序回路は簡単にテストできる．すなわち，k - UCP 順序回路の組合せ部分への基本系列の印加は簡単にできる．以下ではまず，その一例を示す．次に，k - UCP 順序回路の一般のテスト方法を示す．以下では，$Si(r)$ と $Ti(r)$ はそれぞれ基本系列 Si と Ti の r ビット目の値を表す．

例 6.2　全可観測な環境で，図 6.4 に示した順序回路は次のようにテストできる．

(1)　$L7$ に 0 を，$L9$ に $T2$ を，$L14$ に $S2$ を，$L15$ に $S3$ を印加し，クロック・パルスを与えることにより，$L7$ と $L8$ 以外のすべての信号線の縮退故障がテストできる．

(2)　$L7$ に 0 を，$L9$ に $T2(1)$ を，$L14$ に $S2(1)$ を，$L15$ に $S3(1)$ を印加し，クロック・パルスを与える．このとき，$S2(1)$ は $L4$ に，$S1(1)$ は $L1$ に現われる．これにより，$S2(1)$ は $L6$ に現われる．つまり，$S1(1)$ が $L8$ に印加される．よって，$S1$ と $S3$ の 1

ビット目の値が $L7$ と $L8$ に印加される．同様にして，残りのビットを印加することができる．このようにして，$L7$ と $L8$ の縮退故障もテストできる．　□

一般の順序回路は次の手続きによりテストされる．

手続き TSE（k-UCP 順序回路をテストする）

(1)　任意のフリップ・フロップ Fi に対して，次の操作を行なう．Fi がI型フリップ・フロップであれば，その k-UCP 組合せ部分回路の外部入力線に対応する基本系列を印加する．Fi の入力線がNAND (NOR) II型フリップ・フロップであれば，0 (1) を制御線に，1 (0) を伝搬線に，P を印加線に印加し，クロック・パルスを与える．ここで，P を Fi を通じて印加すべき基本系列とする．また，残りの外部入力線に対応する基本系列を印加する．

(2-a)　$r=1$ とする．

(2-b)　NAND (NOR) II型フリップ・フロップに対して，0 (1) を制御線に，1 (0) を伝搬線に，$P(r)$ を印加線に印加し，クロック・パルスを与える．ここで，P を Fi を通じて印加すべき基本系列とする．また，残りの外部入力線に対応する基本系列の r ビット目の値を印加する．

(2-c)　$r=r+1$ とする．$r \leqq k+1$ であれば，(2-a) へ戻る．　□

手続き TSE に示したように，一般の k-UCP 順序回路のテストは2段階に分けて行なわれる．第1段階では，$k+1$ 個のテストベクトルが印加される．第2段階では，$2(k+1)$ 個のテストベクトルが印加される．第8章で述べるように，それらのテストベクトルで縮退故障の故障診断を行なうこともできる．したがって，次の定理が成り立つ．

定理 6.1　全可観測な環境において，$3(k+1)$ 個のテストベクトルで k-UCP 順序回路のすべての縮退故障の検出と診断を行なうことができる．また，それらのテストベクトルを連続的に繰り返し印加することができる．

k は通常2または3なので，全可観測な環境においては，k-UCP 順序回路の故障検出と故障診断は，少ないテストベクトルで行なうことができる．さらに，$3(k+1)$ 個のテストベクトルを連続的に繰り返し印加することができるので，これらのテストベクトルは電子ビームテスタを用いるテスト環境に適用できる．

6.4　k-UCP スキャン回路

スキャンパスをもつ順序回路の場合，そのスキャンパスを再構成することにより，組合せ

部分への基本系列の印加を簡単にすることができる．以下ではまず，基本系列の性質について述べる．次に，k-UCPスキャン回路の概念とテスト方法について述べる．

6.4.1　基本系列の性質

定義6.2　$A = a1\ a2 \ldots an$ と $B = b1\ b2 \ldots bn$ を論理値0と1からなる系列とする．$a1 = bn$ と $ai = bi\text{-}1$ $(i = 2, \ldots, n)$ であれば，A を B の循環シフトと呼び，$A = r(B)$ で表す．

例えば，$A = 10010$ は $B = 01001$ の循環シフトである．縮退故障の基本系列の間にも循環シフトの関係が存在する．縮退故障の2-基本系列では，$S1 = r(S3), S3 = r(S2), T2 = r(T1)$ などが簡単に確かめられる．一般に，1個の基本系列が他の基本系列の循環シフトであるための必要十分条件を次に示す．

補題6.1　k-基本系列 $Si\,(Ti)$ が $Sj\,(Tj)$ の循環シフトである必要十分条件は $i = j + 1 \bmod k + 1$ $(i, j = 1, 2, \ldots, k+1)$ である．

（証明）　T基本系列は S基本系列の否定なので，ここでは，S基本系列の場合のみについて証明する．

まず十分性を証明する．i と j が $i = j + 1 \bmod k + 1$ $(i, j = 1, 2, \ldots, k+1)$ を満たすとする．i＞1のとき，$j = i - 1$ となる．Si の i ビット目の値が0で，他は全部1なので，$Si = r(Sj\text{-}1)$ である．$i = 1$ のとき，$j = k + 1$ なので，$S1 = r(Sk{+}1)$ である．

次に必要性を証明する．$Si = r(Sj)$ とする．i＞1のとき，Si の定義から明らかなように，j は必ず i-1である．また，$i = 1$ のとき，j は必ず $k+1$ である．　□

定義6.3　$h1\text{-}h2\text{-}\ldots\text{-}hn$ を $1, 2, \ldots, k+1$ からなる系列とし，Si と Ti $(i = 1, 2, \ldots, k+1)$ を k-基本系列とする．$Shi = r(Shi{+}1)$ $(i = 1, 2, \ldots, n-1)$ であれば，$h1\text{-}h2\text{-}\ldots\text{-}hn$ を k-有効系列と呼ぶ．

例えば，1-3-2, 3-2-1, 2-1-3は2-有効系列である．しかし，1-2-3は2-有効系列ではない．k-有効系列の定義から，次の補題は明らかである．

補題6.2　$h1\text{-}h2\text{-}\ldots\text{-}hm\ \text{-}\ c$，$c\text{-}f1\text{-}f2\text{-}\ldots\text{-}fn$ が k-有効系列であれば，$h1\text{-}h2\text{-}\ldots\text{-}hm\ \text{-}\ c\text{-}f1\text{-}f2\text{-}\ldots\text{-}fn$ も k-有効系列である（$m \geqq 2$，$n \geqq 2$）．

補題6.3　$h1\text{-}h2\text{-}\ldots\text{-}hi\text{-}hi{+}1\text{-}\ldots\text{-}hm$ が k-有効系列であれば，$h1\text{-}h2\text{-}\ldots\text{-}hi$ も k-有効系列である（$2 \leqq i \leqq m$）．

例えば，2-1-3，3-2-1，1-3-2が2-有効系列なので，補題6.2より，2-1-3-2-1-3-2も2-有効系列である．また，補題6.3より，2-1-3-2も2-有効系列である．

i-(i-1)-...-2-1-...-(k+1)-k-...-(i+2)-(i+1)を任意回繰り返して得られる系列を$Ek(i)$で表す（i=1,2,...,k+1）．例えば，$E2(1)$=1-3-2-1-3-2-...である．k-有効系列の定義と補題6.2より，$Ek(i)$がk-有効系列であることが分かる．

$Ek(i)$の1番目の要素からm番目の要素までの部分系列を$Ek(i, m)$で表す．例えば，$E2(1, 5)$=1-3-2-1-3である．k-有効系列の定義と任意のk-有効系列がi（i=1,2,...,k+1）で始まることより，次の補題が成り立つ．

補題6.4　iで始まる長さmのk-有効系列は$Ek(i, m)$で表すことができる．

例えば，2-1-3 =$E2(2, 3)$，3-2-1-3-2-1-3 =$E2(3, 7)$である．

k-有効系列に関するもう一つの問題は，$h1, h2, ..., hm$（hi=1,2,...,k+1; i=1,2,...,m）が長さmの基本系列を構成するか否かである．$h1, h2, ..., hm$の中のt（t=1,2,...,k+1）の数を$Qt(h1, h2, ..., hm)$で表す．例えば，$Q1(1, 2, 1, 1, 3)$=3，$Q2(1, 2, 1, 1, 3)$=1，$Q3(1, 2, 1, 1, 3)$=1である．明らかに，次の補題が成り立つ．

補題6.5　$h1, h2, ..., hm$（hi=1,2,...,k+1; i=1,2,...,m）が長さmのk-有効系列を構成するための必要十分条件はi（i=1,2,...,k+1）が存在し，$Qt(Ek(i, m))$ =$Qt(h1, h2, ..., hm)$（i=1,2,...,k+1）を満たすことである．また，そのk-有効系列は$Ek(i, m)$である．

例えば，1,1,2,2,3を考える．$Q1(E2(2, 5))$=2=$Q1(1, 1, 2, 2, 3)$，$Q2(E2(2, 5))$=2=$Q2(1, 1, 2, 2, 3)$，$Q3(E2(2, 5))$=1=$Q3(1, 1, 2, 2, 3)$であるので，1,1,2,2,3は2-有効系列を構成する．その系列は2-1-3-2-1 =$E2(2, 5)$である．

6.4.2　基本概念

まず，k-UCPスキャン回路の一例を示す．

例6.3　図6.5に示す順序回路を考える．その組合せ部分がすでに2-UCP回路に変換されている．また，各フリップ・フロップの出力線に割り当てられた極性はすべて正極性である．SI-$F3$-$F2$-$F1$-SOのようにスキャンパスが構成されている．$F1, F2, F3$の出力線に割り当てられた色は$C1, C2, C3$である．また，1-3-2は2-有効系列である．

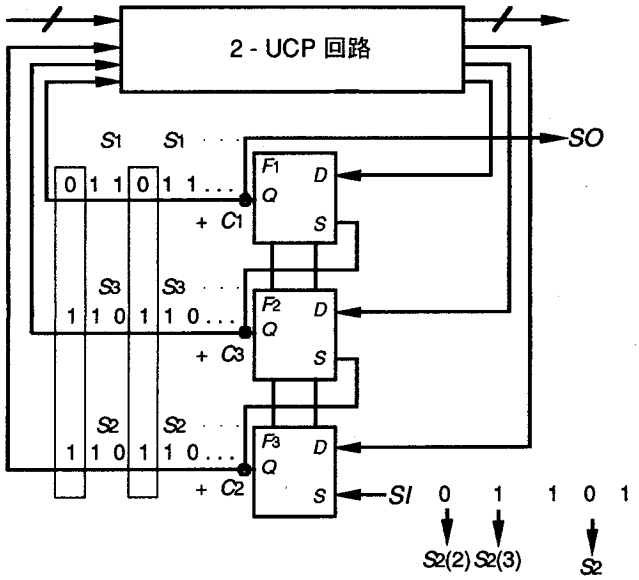


図 6.5　基本系列の印加例

まず，$F1, F2, F3$ に 0, 1, 1 を設定する．次に，$0 = S2(2), 1 = S2(3), S2 \ldots$ を SI より順次に印加する．ここで，$S2(2)$ は基本系列 $S2$ の 2 ビット目の値を表す．明らかに，$F1, F2, F3$ の出力はそれぞれ $S1, S3, S2$ の繰り返しである．すなわち，この順序回路の 2 - UCP 組合せ部分に所要の基本系列を連続的に繰り返し印加することができる．これにより組合せ部分を短い周期の周期動作にすることができる．□

図 6.5 に示した順序回路の組合せ部分には，基本系列を連続的に繰り返し印加することができる．一般に，このような性質をもつ回路を次のように定義することができる．

定義 6.4　図 6.6 の $F1, F2, \ldots, Fn$ をフリップ・フロップとする．スキャンパス $SI \rightarrow Fn \rightarrow Fn\text{-}1 \rightarrow \ldots \rightarrow F1 \rightarrow SO$ をもつ順序回路が次の条件を満たせば，この順序回路を k - UCP スキャン回路と呼ぶ．

(1)　組合せ部分が k - UCP 回路である．

(2)　$F1, F2, \ldots, Fn$ の出力線が同じ極性をもつ．

(3)　$F1, F2, \ldots, Fn$ の出力線の色を $Ch1, Ch2, \ldots, Chn$ とすると，$h1 - h2 - \ldots - hn$ が k - 有効系列である．

明らかに，図 6.5 に示した回路は 2 - UCP スキャン回路である．

6.4.3　テスト方法

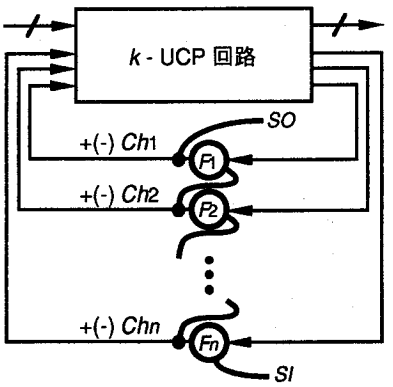


図 6.6　k - UCP スキャン回路の定義

一般に，k - UCP スキャン回路は次の手続きによりテストできる．

手続き TSC（k - UCP スキャン回路をテストする）

(1)　すべてのフリップ・フロップの出力線が正極性をもつとき，スキャンパスを通じて $F1, F2, \ldots, Fn$ に $Sh1(1), Sh2(1), \ldots, Shn(1)$ を印加する．すべての フリップ・フロップの出力線が負極性をもつとき，スキャンパスを通じて $F1, F2, \ldots, Fn$ に $Th1(1), Th2(1), \ldots, Thn(1)$ を印加する．

(2)　すべてのフリップ・フロップが正極性をもつとき，SI に $Sh2(2), Sh2(3), \ldots, Sh2(k+1)$ を印加する．すべてのフリップ・フロップ が負極性をもつとき，SI に $Th2(2), Th2(3), \ldots, Th2(k+1)$ を印加する．　□

定理 6.2　全可観測な環境において，$k+1$ 個のテストベクトルで k - UCP スキャン回路のすべての縮退故障の故障検査と故障診断を行なうことができる．また，それらのテストベクトルを連続的に繰り返し印加することができる．

（証明）　ここでは，すべてのフリップ・フロップが正極性をもつ場合について証明する．すべてのフリップ・フロップが負極性をもつ場合については，同様に証明できる．

k - UCP スキャン回路においてまず，スキャンパスを通じて，$F1, F2, \ldots, Fn$ に $Sh1(1), Sh2(1), \ldots, Shn(1)$ を印加する．次に，SI に $Sh2(2), Sh2(3), \ldots, Sh2(k+1)$ を印加する．ここで，$Si(t)$ は基本系列の Si の t ビット目の値を表す．これにより，Fn の出力は Shn となる．Ft の出力を Sht （$t \leq n$）とすると，Ft-1 の出力は Sht-1(1), $Sht(1), Sht(2), \ldots, Sht(k)$ となる．$h1 - h2 - \ldots - hn$ が k - 有効系列なので，Sht-1 = Sht である．つまり，Ft-1 の出力は Sht-1 である．したがって，Fi の出力は必ず Shi である．

SI にさらに $Shn(1)$ を印加したとき，$F1, F2, \ldots, Fn$ にそれぞれ $Sh2(k+1), Sh3(k+1), \ldots, Shn-1(k+1), Shn(1)$ が現れる．$h1 - h2 - \ldots - hn$ が k - 有効系列なので，$Sh1(k+1) = Sh1(1)$,

$Sh3(k+1) = Sh2(1), \ldots, Shn\text{-}1(k+1) = Shn\text{-}1(1)$ となる．つまり，SI にさらに $Shn(2), Shn(3), \ldots, Shn(k+1)$ を順次印加すると，Fi の出力は必ず Shi の繰り返しとなる．また，SI に Shn を繰り返し印加すると，Fi の出力は必ず Shi の繰り返しとなる．□

6.5　まとめ

全可観測な環境においては，順序回路を少ないテストベクトルでテストするため，その組合せ部分を k-UCP 回路に変換することが考えられる．組合せ部分が k-UCP 回路である順序回路のすべての縮退故障のテストは，本来 $k+1$ 個のテストベクトルで十分である．しかし，その $k+1$ 個のテストベクトルを印加することは，フリップ・フロップの出力線を経由しなければならない場合，困難かまたは不可能である．この問題を解決するため，全可観測な環境でのテスト容易な順序回路を提案した．

まず，フリップ・フロップの入力側の構造を限定することによりテストベクトルの印加を容易にする k-UCP 順序回路を提案し，k-UCP 順序回路内のすべての縮退故障は $3(k+1)$ 個のテストベクトルでテストできることを示した．$3(k+1)$ 個のテストベクトルの内の $2(k+1)$ 個は本来必要な $k+1$ 個のテストベクトルを印加するためのものである．順序回路がスキャンパスをもつ場合，そのスキャンパスの組み方を工夫することにより，$k+1$ 個のテストベクトルを余分な入力ベクトルを伴わずに印加することができる．そのような順序回路は提案した k-UCP スキャン回路である．つまり，k-UCP スキャン回路内のすべての縮退故障は $k+1$ 個のテストベクトルでテストできる．

組合せ回路と異なり，順序回路の場合は，回路の短いテスト系列を繰り返し印加することができない可能性がある．そのとき，回路の短い周期の周期動作を確立することができないので，電子ビームテスタを利用した故障診断はできない．提案したテスト容易な順序回路においては，短いテスト系列を回路に連続的に繰り返し印加することができる．このため，提案した回路は電子ビームテスタを用いた故障診断に適していることが分かる．

7 全可観測な環境での順序回路のテスト容易化設計

第6章で述べたように，k-UCP順序回路またはk-UCPスキャン回路の縮退故障については，全可観測な環境において，少ないテストベクトルで検出と診断を行なうことができる．しかし，任意の順序回路は，必ずしもk-UCP順序回路またはk-UCPスキャン回路であるとは限らない．本章では，任意の順序回路をk-UCP順序回路またはk-UCPスキャン回路に変換する方法を提案する [52, 53].

7.1 k-UCP順序回路への変換

7.1.1 回路変換手法

ここで，変換前の順序回路はn個のフリップ・フロップ$F1, F2, \ldots, Fn$をもつとする．

手続き SM1 （k-UCP順序回路への変換を行なう）

(1) 組合せ部分をk-UCP回路に変換する．

(2) $t=1$とする．

(3) FtがI型フリップ・フロップであれば，(6)へ移る．

(4) Ftの入力線があるゲートの出力線であれば，その入力線をLとする．そうでなければ，Ftの入力線が必ず分岐の枝である．このとき，分岐の幹をLとする．

(5) Lが正（負）極性をもつなら，図6.3に示したように直列につなぐ2個のk入力NAND（NOR）ゲートをLへ挿入する．

(6) $t=t+1$とする．$t \leq n$であれば，(3)へ戻る． □

例えば，図6.2に示した順序回路は手続きSM1により，図6.4に示した2-UCP順序回路に変換されている．

7.1.2　実験結果

手続き SM1 に示したように，任意の順序回路は 2 段階を経て k-UCP 順序回路に変換される．第 1 段階では，順序回路の組合せ部分が k-UCP 回路に変換される．第 2 段階では，フリップ・フロップの入力側に必要に応じて入力ゲートを挿入する．第 2 段階では，最良の場合には，付加ゲート数は 0 であるが，最悪の場合には，付加ゲート数は $2n$ である．なお，n は順序回路内のフリップ・フロップの数である．また，第 1 段階と第 2 段階での付加外部入力線数はそれぞれ $k+1$ と $2(k-1)$ である．

C 言語で手続き SM1 をプログラム化し，実験を行なった．使用したコンピュータは富士通の S-4/LC（12.5 MIPS）である．実験用の回路は ISCAS 1989 のベンチマーク順序回路である [72].

実験結果を表 7.1 に示す．"original Tr. s" は変換前の回路のトランジスタの数を，"2-U" は組合せ部分が 2 入力 NAND（NOR）ゲートで構成されるように変換するために付加したトランジスタの数を，"color" は組合せ部分が彩色解をもつように変換するために付加したトランジスタの数を，"IG" は入力ゲートを挿入するために付加したトランジスタの数を表す．

表 7.1　k-UCP 順序回路への変換の実験結果

circuit	original Tr.s (): F/F	added Tr. s			overhead (%)		time (sec.)
		2-U	color	IG	whole	partial	
s27 (c)	96 (3)	10 (NOR)	0	16	27.1	15.1	0.6
s208 (c)	518 (8)	168 (NOR)	24	36	44.0	8.7	9.4
s298 (c)	834 (14)	336 (NOR)	80	112	63.3	16.4	16.4
s344 (c)	914 (15)	98 (NOR)	128	84	33.9	20.9	19.4
s349 (c)	924 (15)	216 (NAND)	40	72	35.5	9.8	20.4
s382 (c)	1060 (21)	436 (NAND)	64	92	55.8	10.4	26.6
s420 (c)	1034 (16)	402 (NAND)	24	88	49.7	7.8	39.4
s444 (c)	1136 (21)	460 (NAND)	48	92	52.8	8.8	26.1
s510 (c)	1082 (6)	564 (NAND)	160	44	71.0	12.4	79.7

"2-U" と "color" の結果は 2-UCP 回路である．結果として，s27, s208, s298, s344 の組合せ部分は 2 入力 NOR ゲートで構成されるように変換された．s349, s382, s420, s444, s510 の組合せ部分は 2 入力 NAND ゲートで構成されるように変換された．

"whole" はベンチマーク順序回路を 2-UCP 順序回路に変換したときのオーバーヘッドを表す．また，"partial" はもとの回路の組合せ部分が 2 入力 NAND または NOR ゲートで構成されると仮定して 2-UCP 順序回路に変換したときのオーバーヘッドを表す．それぞれは次のように計算した．

$$\text{whole} = (\text{2-U}) + \text{color} + \text{IG}) \,/\, (\text{original Tr.s})$$

$$\text{partial} = (\text{color} + \text{IG}) \,/\, (\text{original Tr.s} + (\text{2-U}))$$

7.2　k-UCP スキャン回路への変換

7.2.1　回路変換手法

k-UCP スキャン回路への回路変換は2段階に分けて行なう．まず，与えられた順序回路の組合せ部分を k-UCP 回路に変換する．次に，回路全体が k-UCP スキャン回路の定義の2番目と3番目の条件を満たすようにさらに変換を行なう．

任意の組合せ回路を k-UCP 回路に変換する手法については，第5章で述べた．

k-UCP 回路に対して，極性の割り当ての結果は1通りしかないので，もし各フリップ・フロップの出力線の極性が異なるならば，付加ゲートで極性を調整する必要がある．図5.4に調整の方法を示した．この調整では，1個のフリップ・フロップに対して，高々1個の k 入力 AND または OR が必要である．

各フリップ・フロップに割り当てられた色を $Ch1, Ch2, \ldots, Chn$ とする．$h1, h2, \ldots, hn$ が $t1 \rightarrow t2 \rightarrow \ldots \rightarrow tn$ という k-有効系列を構成すれば，$SI \rightarrow Ftn \rightarrow Ftn\text{-}1 \rightarrow \ldots \rightarrow Ft1 \rightarrow SO$ というスキャンパスを構成することにより k-UCP スキャン回路が得られる．$h1, h2, \ldots, hn$ が k-有効系列を構成しなければ，次に述べる変換が必要である．

k-UCP 回路に対して，色の割り当ての結果が1通り以上ある場合もある．その場合に，考えられるのはすべての可能な色の割り当てを求め，その中に k-有効系列に対応する色の割り当てが存在するかどうかを調べることである．しかし，すべての彩色解を求めることは時間がかかる．以下では，ある彩色解が与えられたとして，回路変換を行なうことにする．

ここで，フリップ・フロップの $F1, F2, \ldots, Fn$ に割り当てられた色を $Ch1, Ch2, \ldots, Chn$ とし，$h1, h2, \ldots, hn$ が k-有効系列を構成しないとする．この場合，付加ゲートを用いて回路変換を行なう必要がある．提案する回路変換手法は次の手続きで与えられる．

手続き MS2（k-UCP スキャン回路への変換を行なう）

(1)　組合せ部分を k-UCP 回路に変換する．その k-UCP 回路の一つの彩色解を求める．

(2)　正極性をもつフリップ・フロップの数を $F+$，負極性をもつフリップ・フロップの数を $F-$ とする．$F+$ と $F-$ を求める．

(3)　$F+ > F-$（$F+ \leq F-$）の場合，AND（OR）ゲートを負極性（正極性）をもつフリップ・フロップの出力線に挿入する．

(4)　Pt を空にする（$t = 1, 2, \ldots, k+1$）．

(5)　$t \leftarrow 1$ とする．

(6)　Wr を空にする（$r = 1, 2, \ldots, k+1$）．

(7)　色 Ci をもつフリップ・フロップを集合 Wi に入れる（$i = 1, 2, \ldots, k+1$）．

(8)　$Ek(t, m)$ の示す順番に集合 Wr（$r = 1, 2, \ldots, k+1$）から要素を取り出し，列 Pt を作

る．もし空のWdから要素を取り出す必要があれば，空でなくかつ残りの要素の一番多い列Wgから要素フリップ・フロップを取り出す．このフリップ・フロップに$<d, g>$というマークを付ける．

(9)　$t \leftarrow t+1$とする．$t \leqq k+1$の場合，(6)に戻る．$t > k+1$の場合，次へ移る．

(10)　フリップ・フロップに付けられたマークが一番少ないPtを選び，それをPとする．

(11)　Pの示す順にスキャンパスを構成する．出力線に正極性（負極性）をもつフリップ・フロップに$<d, g>$マークが付けられていれば，2個のk入力NAND（NOR）ゲートをフリップ・フロップの出力線に挿入して，出力線の色をCgからCdに調整する．□

(2)と(3)では，各フリップ・フロップの出力線が同じ極性をもつように調整する．(4)～(11)では，なるべく少ない付加ゲートで回路全体をk-UCPスキャン回路に変換する．次は回路変換手続きSM2の実行例である．

例7.1　ある順序回路の組合せ部分がすでに2-UCP回路に変換されたとする．各フリップ・フロップの出力線に割り当てられた極性と色を図7.1に示す．$F2$のみが負極性をもつので，$F+$と$F-$はそれぞれ7と1である．1個の2入力ANDゲートを図7.2に示すように$F2$の出力線に挿入することにより，極性の調整を行なう．

各フリップ・フロップの出力線に割り当てられた色は$C1, C1, C2, C2, C3, C3, C3, C3$である．補題6.5から，1, 1, 2, 2, 3, 3, 3, 3が2-有効系列を構成しないことが分かる．

$W1, W2, W3$は次のようになる．

$W1$：　$F1, F2.$
$W2$：　$F3, F4.$
$W3$：　$F5, F6, F7, F8.$

$t=1$のとき，$E2(1, 8) = 1-3-2-1-3-2-1-3$が使われる．このとき，$P1 = F1-F5-F3-F2-F6-F4-F8<1, 3>-F7$が求められる．同様に，$t=2$と$t=3$のとき，$P3 = F3-F1-F5-F4-F2-F6-F8<2, 3>-F7<2, 3>$，$P3 = F5-F3-F1-F6-F4-F2-F7-F8<1, 3>$がそれぞれ求められる．この場合，$P$に$P1$を選べばよい．$P$の示す順に形成されたスキャンパスは図7.3に示す．2個の2入力NANDゲートが$F8$の色を$C3$から$C1$に調整するために使われる．□

7.2.2　実験結果

任意の順序回路をk-UCPスキャン回路に変換するためのオーバーヘッドは，順序回路

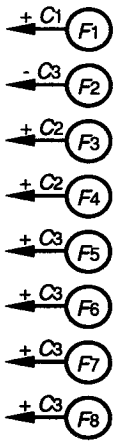


図 7.1　変換前のフリップ・フロップ

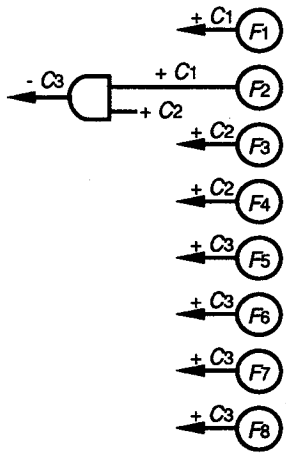


図 7.2　極性調整後のフリップ・フロップ

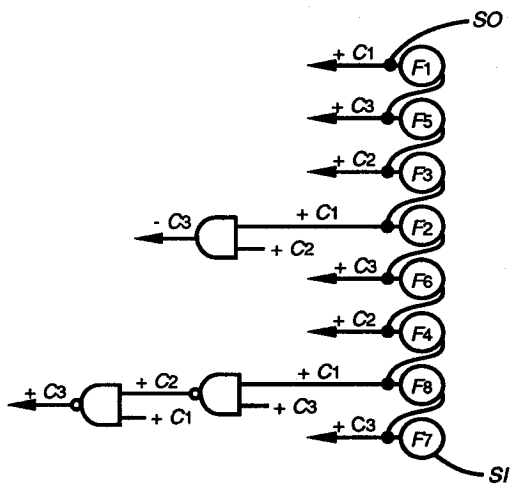


図 7.3　形成されたスキャンパス

の組合せ部分を k - UCP 回路に変換するためのオーバーヘッドとスキャンパス部分についての調整に必要なオーバヘッドを含む．後者は1個のフリップ・フロップに対して，最悪の場合3個，最良の場合0個の k 入力の付加ゲートを必要とする．また，付加外部入力線の数は $k+1$ である．

C言語で手続き SM2 をプログラム化し，実験を行なった．使用したコンピュータは富士通の S-4/LC（12.5 MIPS）である．実験用の回路は ISCAS 1989 のベンチマーク順序回路である [72].

実験結果は表7.2に示す．"original Tr. s" は変換前の回路のトランジスタの数を，"2-U" は組合せ部分が2入力 NAND（NOR）ゲートで構成されるように変換するために付加したトランジスタの数を，"color" は組合せ部分が彩色解をもつように変換するために付加したトランジスタの数を，"P/C" は k - UCP スキャン回路の定義の2番目と3番目の条件を満足させるために付加したトランジスタの数をを表す．

"2-U" と "color" の結果は2 - UCP 回路である．s27, s208, s298, s344 の組合せ部分は2入力 NOR ゲートで構成されるように変換された．s349, s382, s420, s444, s510 の組合せ部分は2入力 NOR ゲートで構成されるように変換された．

"whole" はベンチマーク順序回路を2 - UCP 順序回路に変換したときのオーバーヘッドを，"partial" はもとの回路の組合せ部分が2入力 NAND または NOR ゲートで構成されると仮定して2 - UCP 順序回路に変換したときのオーバーヘッドを表す．それぞれは次のように計算した．

$$\text{whole} = (\text{2-U}) + \text{color} + \text{P/C}) \,/\, (\text{original Tr.s})$$

$$\text{partial} = (\text{color} + \text{P/C}) \,/\, (\text{original Tr.s} + (\text{2-U}))$$

表7.2　k - UCP スキャン回路への変換の実験結果

circuit	original Tr.s (): F/F	added Tr. s			overhead (%)		time (sec.)
		2-U	color	P/C	whole	partial	
s27 (cs)	96 (3)	10 (NOR)	0	8	10.3	4.3	1.0
s208 (cs)	518 (8)	168 (NOR)	24	32	30.9	6.3	9.1
s298 (cs)	834 (14)	336 (NOR)	80	32	37.4	7.3	16.3
s344 (cs)	914 (15)	98 (NOR)	128	72	22.9	14.3	20.2
s349 (cs)	924 (15)	216 (NAND)	40	72	25.0	7.3	20.7
s382 (cs)	1060 (21)	436 (NAND)	64	112	38.1	8.6	26.6
s420 (cs)	1034 (16)	402 (NAND)	24	48	32.7	3.9	36.1
s444 (cs)	1136 (21)	460 (NAND)	48	88	35.4	6.3	26.1
s510 (cs)	1082 (6)	564 (NAND)	160	24	60.4	10.2	77.8

実験結果から，次のような考察を行なうことができる．

(1)　表7.2の "whole" と "partial" の値は表7.1のそれより小さい．つまり，もとの順序

回路がスキャンパスをもつならば，それを k-UCP スキャン回路に変換するためのオーバーヘッドは小さい．

(2) 表7.1と表7.2では，"partial" の値が "whole" の値より小さい．つまり，与えられた順序回路の組合せ部分が k 入力 NAND または NOR ゲートで構成されている場合，それを k-UCP 順序回路，または k-UCP スキャン回路に変換するためのオーバーヘッドは小さい．

(3) 表7.2の "partial" の値が示したオーバーヘッドの中で一番小さい．つまり，もとの順序回路がスキャンパスをもち，かつその組合せ部分が k 入力 NAND または NOR ゲートで構成される場合，それを k-UCP スキャン回路に変換するためのオーバーヘッドは小さい．

7.3　まとめ

本章では，任意の順序回路を k-UCP 順序回路に変換する手法，および k-UCP スキャン回路に変換する方法を提案した．これらの変換では，まず最初に順序回路の組合せ部分を k-UCP 回路に変換する．これについて，第5章ですでに述べた．本章では，順序回路のフリップ・フロップ周りに関する調整を行ない，与えられた順序回路をテスト容易な順序回路に最終的に変換する手法を提案した．

k-UCP 順序回路に変換するとき，フリップ・フロップの入力線へ必要に応じて入力ゲートを挿入するだけでよい．k-UCP スキャン回路に変換するとき，フリップ・フロップのすべての出力線の極性が同じで，色の番号が k-有効系列を構成するように付加ゲートで調整しなければならない．本章では，これをなるべく少ない付加ゲートで行なうための手法を提案した．

同じベンチマーク回路に対して回路変換の実験を行った結果，k-UCP 順序回路に変換したときの平均オーバーヘッドは 48.12 % であり，k-UCP スキャン回路に変換したときの平均オーバーヘッドは 32.56 % である．また，与えられた回路の組合せ部分が k 入力 NAND または NOR ゲートのみからなる場合，k-UCP 順序回路に変換したときの平均オーバーヘッドは 12.26 % であり，k-UCP スキャン回路に変換したときの平均オーバーヘッドは 7.61 % である．この結果より，もとの順序回路がスキャンパスをもち，かつその組合せ部分が k 入力 NAND または NOR ゲートで構成されている場合，それを k-UCP スキャン回路に変換するためのオーバーヘッドが小さく実用範囲内であることが分かった．

8 全可観測な環境での論理回路の故障診断

本章ではまず，CMOS の k - UCP 回路のスタック・オープン故障の診断手法について述べる [38, 39]．k - UCP 回路の特徴を利用して効率的な故障診断ができることを示す．次に，一般の論理回路の故障診断にも用いられるガイディド・プローブ法の効率化について述べる [61].

8.1　k - UCP 回路の故障診断

8.1.1　基本原理

4.3 に示した手続き TG により生成されたテスト集合を k - UCP 回路に印加したとき，故障ゲートの出力系列は基本系列ではない．例えば，図8.1 に示すように，スタック・オープン故障用の 2 - 基本系列 $S1$ と $S2$ が CMOS の 2 入力 NAND ゲートに印加されたとする．正常時の出力系列は $S3$ である．図 2.5 に示した 2 入力 NAND ゲートの CMOS の構成では，トランジスタ Ta にスタック・オープン故障があるとする．このとき，出力系列は $S3$ ではなく，$S3^*(2) = 0011010$ となる．一般に，$Si^*(b1, b2, \ldots, bm)$（$Ti^*(b1, b2, \ldots, bm)$）は $b1$, $b2, \ldots, bm$ ビット位置で Si（Ti）と異なる系列を表す．

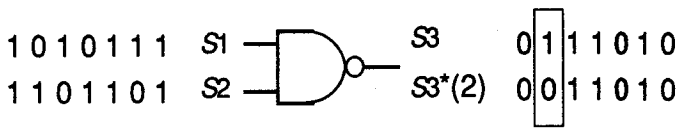


図 8.1　2 入力 CMOS の NAND ゲートへの入力

同様に，$T1$ と $T2$ を CMOS の 2 入力 NOR ゲート に印加した場合，正常時のゲートの出力系列は $T3$ である．図 2.6 に示した 2 入力 NOR ゲートの CMOS の構成では，トランジスタ Ta にスタック・オープン故障があるとする．このとき，出力系列は $T3$ ではなく，$T3^*(2) = 1100101$ となる．

定義 8.1　テスト集合が故障のある k-UCP 回路に印加されたとき，故障ゲートの出力系列を故障系列と呼ぶ．

以下では，図 2.11 と図 2.12 に示した CMOS の k 入力 NAND (NOR) ゲートのトランジスタ Pi (Ni)，および CMOS の k 入力 AND (OR) ゲートのトランジスタ Pi (Ni) と N (P) を 1 個のトランジスタと見なす．これらのトランジスタをトランジスタ g (g = 1, 2, . . ., k) で表す．CMOS の k 入力 NAND (NOR) ゲートのトランジスタ $N1, N2, \ldots, Nk$ ($P1, P2, \ldots, Pk$)，および CMOS の k 入力 AND (OR) ゲートのトランジスタ $N1, N2, \ldots, Nk$ ($P1, P2, \ldots, Pk$) と P (N) を 1 個のトランジスタと見なす．これらのトランジスタをトランジスタ NP で表す．また，CMOS の NOT ゲートの p トランジスタを P，n トランジスタを N で表す．

以下では，$I1 = St1, I2 = St2, \ldots, Ik = Stk$ ($I1 = Tt1, I2 = Tt2, \ldots, Ik = Ttk$) を入力とする図 2.11 と図 2.12 に示した k 入力ゲートのトランジスタ g のスタック・オープン故障を $FSt1t2...tk(g)$ ($FTt1t2...tk(g)$) で表し，基本系列 Stf (Ttf) を入力とする NOT ゲートのトランジスタ i のスタック・オープン故障を $FStf(i)$ ($FTtf(i)$) で表す (g = 1, 2, . . ., k, NP; i = P, N)．例えば，$S1$ と $S2$ が 2 入力 NAND ゲートに印加されたとき，トランジスタ 1 のスタック・オープン故障を $FS12(1)$ で表し，$T1$ と $T2$ が 2 入力 NOR ゲートに印加されたとき，トランジスタ 1 のスタック・オープン故障を $FT12(1)$ で表す．

故障ゲートの出力線が他のゲートの入力線でもあるとき，故障系列と基本系列の論理演算が行なわれ，その結果が新しい系列になる可能性がある．例えば，$S4*(2)$ が故障系列であれば，NAND($S4*(2), S1, S3$) = $S1*(2)$ という演算が可能である．

定義 8.2　テスト集合が故障のある k-UCP 回路に印加されたとき，故障のないゲートの出力系列が基本系列でなければ，その出力系列を誤り系列と呼ぶ．

故障系列と誤り系列の論理演算が行なわれる可能性もある．その結果は基本系列または誤り系列である．次にその例を示す．ここでは，$S4*(2)$ は故障系列であり，$S1*(2)$ と $S2*(2)$ は誤り系列である．

$$\mathrm{NAND}(S4*(2), S2, S3) = S1*(2)$$
$$\mathrm{NAND}(S1*(2), S2, S4) = S3*(2)$$
$$\mathrm{NAND}(S2*(2), S1, S3) = S4$$

定義 8.3　故障系列と誤り系列をあわせて異常系列と呼ぶ．

定義 8.4　故障のある k-UCP 回路において，異常系列が現われる信号線をその故障の診

断点と呼び，ある故障のすべての診断点を含む集合をその故障の診断点集合と呼ぶ．

定義 8.5　異常系列において，基本系列と異なるビットの位置を故障の誤りビット位置と呼ぶ．

例 8.1　図4.1に示した2-UCP回路のゲート A のトランジスタ P にスタック・オープン故障（$FT3(P)$）があるとする．この場合，故障系列が $S3^*(2, 6)$ で，誤り系列は $S1^*(2)$ と $S2^*(2)$ である．つまり，次の演算がなされる．

$$\mathrm{NAND}(S3^*(2, 6), S2) = S1^*(2)$$
$$\mathrm{NAND}(S3^*(2, 6), S2) = S2^*(6)$$

□

この例から分かるように，NOTゲート内の故障の誤りビット位置が伝搬により変わる可能性がある．この例では，NOTゲート内の $FT3(P)$ 故障の故障系列の誤りビット位置が2と6であるが，誤り系列の誤りビット位置が2, 6または2と6である．しかし，AND, OR, NAND, NORゲートに関して，次の補題が成り立つ．

補題 8.1　故障ゲートがAND, OR, NAND, NORゲートである場合，誤りビットの位置は伝搬により変わらない．

（証明）　ここで，故障ゲートがCMOSの k 入力NANDゲート G の場合について証明する．他の場合については，ほぼ同様に証明できる．

ゲート G が p トランジスタのスタック・オープン故障をもつ場合，誤りビットが1個しかない．よって，補題8.1が成り立つ．ゲート G が n トランジスタのスタック・オープン故障をもつ場合，補題4.1から，故障系列が k 個の誤りビットをもつ．それらの誤りビットを $b1, b2, \ldots, bk$ とする．補題4.1から，Si と Si の $b1, b2, \ldots, bk$ ビットの値が等しい．また，故障系列の $b1, b2, \ldots, bk$ ビットの値も等しい．したがって，回路内の任意のゲートの出力系列は基本系列または $b1, b2, \ldots, bk$ ビットで基本系列と異なる異常系列である．よって，この補題が成り立つ．□

次の定理は提案するCMOSの k-UCP回路におけるスタック・オープン故障の故障診断法の基礎を与える．

定理 8.1　誤りビット位置と診断点集合により，k-UCP回路内の任意の2個のスタック・オープン故障は区別できる．

（証明）　2個の故障が同じゲートにあれば，補題4.4から，これらの故障は必ず異なる誤りビット位置をもつ．よって，この2個の故障は区別できる．2個の故障が異なるゲートにあれば，これらの故障の誤りビット位置は同じである可能性がある．この場合，この2個の故障の診断点集合は必ず異なる．これは故障ゲートの出力線が必ずその故障の診断点集合に属するからである．したがって，診断点集合により，この2個の故障を区別することができる．よって，この補題が成り立つ．　□

図4.1の回路のすべての可能な故障の誤りビット位置と診断点を表8.1のような表にまとめることができる．このような表を故障診断表と呼ぶ．定理8.1から分かるように，故障診断表は故障位置を特定するために必要なすべての情報が含まれている．例えば，表8.1に示した故障診断表においては，ゲート A は2個の可能な故障 A-1 と A-2 をもつ．故障 A-1 は2と6を誤りビット位置とする診断点 $L6$，6を誤りビット位置とする診断点 $L9$，2を誤りビット位置とする診断点 $L10$, $L16$, $L20$, $L17$ をもつ．

表 8.1　故障診断表の例

		$L6$	$L9$	$L10$	$L14$	$L16$	$L20$	$L17$	$L18$
A-1	$FT3(P)$	2, 6	6	2		2	2	2	
A-2	$FT3(N)$	5, 7	5, 7	5, 7		5, 7	5, 7		5, 7
B-1	$FS13(1)$		4						
B-2	$FS13(2)$		7						
B-3	$FS13(NP)$		3, 6						
C-1	$FS23(1)$			3					
C-2	$FS23(2)$			5		5	5		5
C-3	$FS23(NP)$			2, 4		2, 4	2, 4	2, 4	2, 4
D-1	$FS13(1)$				4				
D-2	$FS13(2)$				7				
D-3	$FS13(NP)$				3, 6	3, 6	3, 6		3, 6
E-1	$FS12(1)$					2			2
E-2	$FS12(2)$					6			
E-3	$FS12(NP)$					5, 7			5, 7
F-1	$FS12(1)$						2	2	2
F-2	$FS12(2)$						6	6	6
F-3	$FS12(NP)$						5, 7		
G-1	$FS23(1)$								3
G-2	$FS23(2)$								5
G-3	$FS23(NP)$								2, 4
H-1	$FT13(1)$							4	4
H-2	$FT13(2)$							7	
H-3	$FT13(NP)$							3, 6	3, 6

8.1.2　故障診断表の生成

故障診断表は故障シミュレーションにより生成できる．ここでは，基本系列と異常系列との論理演算における性質を利用して故障シミュレーションを高速化する手法を提案する．

テスト集合が印加された k-UCP 回路においては，入力線に異常系列があるにもかかわらず，出力線に基本系列があるようなゲートが存在しうる．例えば，次の演算が可能である．

NAND($S4$*(2), $S1$, $S3$) = $S2$

定義 8.6　テスト集合が故障のある k-UCP 回路に印加されたとき，入力線に異常系列があるにもかかわらず，出力線に基本系列があるようなゲートをその故障の伝搬阻止ゲートと呼ぶ．

明らかに，もし NOT ゲートの入力線に異常系列があれば，その出力系列は必ず異常系列である．すなわち，k-UCP 回路内の可能な伝搬阻止ゲートは k 入力 AND, OR, NAND, および NOR ゲートである．次の定理は k 入力ゲート（AND, OR, NAND, NOR）が伝搬阻止ゲートであるか否かを調べる方法を与える．

定理 8.2　テスト集合が故障のある k - UCP 回路に印加されたとき，k 入力ゲート G（AND, OR, NAND, NOR）が伝搬阻止ゲートであるための必要十分条件は以下である．ここで，$t1\ t2 \ldots tk\ tk{+}1$ は $1, 2, \ldots, k+1$ の順列である．

(1)　ゲート G の故障が $FSt1t2...tk(i)$ または $FTt1t2...tk(i)$ である場合，ある入力線に Sti, Sti*, Tti または Tti*があり，かつ少なくとも他の 1 本の入力線に異常系列があれば，ゲート G は伝搬阻止ゲートである．

(2)　ゲート G の故障が $FSt1t2...tk(NP)$ または $FTt1t2...tk(NP)$ である場合，ある入力線に $Stk{+}1$, $Stk{+}1$*, $Ttk{+}1$ または T$tk{+}1$*があり，かつ少なくとも他の 1 本の入力線に異常系列があれば，ゲート G は伝搬阻止ゲートである．

（証明）　ここで，k 入力 NAND ゲートと $FSt1t2...tk(1)$ 故障の場合について証明する．他の場合については，ほぼ同様に証明できる．

補題4.4から分かるように，故障ゲートに印加される図8.2に示すようなベクトルの対は1個しかない．明らかに，$S1$（$S1$*）の第 i ビットの値は 0 (1) であり，$S2, \ldots, Sk{+}1$ ($S2$*, ..., $Sk{+}1$*) の第 i ビットの値は 0 (1) である．つまり，この故障の誤りビットは第 i ビットである．

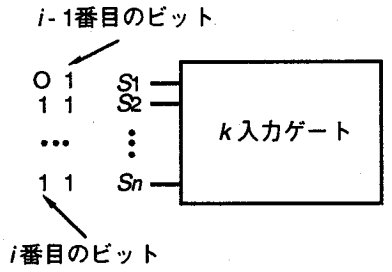


図 8.2　定理 8.2 の証明

まず，十分条件を証明する．$S1$ または $S1^*$ がゲート G に印加されたとする．そのゲートの出力系列の第 i ビットの正常値は 1 である．もし少なくとも他の 1 本の入力線に異常系列があれば，そのゲートの出力系列の第 i ビットの値は必ず 1 である．それは，その異常系列の第 i ビットの値が必ず 0 であるからである．よって，出力系列は基本系列である．

次に背理法を用いて必要条件を証明する．ゲート G を伝搬阻止ゲートとし，異常系列として $S1^*$ のみが印加されたとする．この場合，出力系列の第 i ビットの値が正常値と異なって 0 となる．つまり，G は伝搬阻止ゲートではないことになる．ゲート G を伝搬阻止ゲートとし，$S1^*$ 以外の異常系列のみが印加されたとする．この場合，出力系列の第 i ビットの正常値は 0 であるが，故障時には 1 となる．それは異常系列の第 i ビットの値が 0 であるからである．よって，G は伝搬阻止ゲートではないことになる．□

定義 8.7　故障ゲートの出力線より外部出力端子に向かって到達可能なゲートを到達可能ゲートと，到達可能ゲートへの入出力信号線を対象信号線と呼ぶ．

定理 8.2 を用いて，k 入力ゲート（AND, OR, NAND, NOR）内の故障の故障診断点集合は，N（正常），E（異常）および X（未知）という信号値でシミュレーションを行なうことにより求めることができる．これを次の手続きに示す．

手続き DTG（故障診断点集合を求める）

(1)　故障ゲートの出力線に E を，到達可能ゲートでないゲートの出力線または外部入力線で対象信号線でもある信号線に N，それ以外の対象信号線に X を設定する．

(2-a)　すべての入力値が N のゲートの出力線に，N を設定する．

(2-b)　伝搬阻止ゲートでないゲートの入力に，E が一つでもあれば，出力線に E を設定する．

(2-c)　伝搬阻止ゲートの出力線に N を設定する．

(3)　すべての対象信号線の X がなくなるまで，(2) を繰り返す．□

手続き DTG により E に設定された信号線がその故障の診断点集合を構成する．

例 8.2　図 4.1 に示した回路のゲート C の故障 $FS23(1)$ の故障診断集合を求める．この故障の到達可能ゲートは E, F, G, H であり，対象信号線は $L11$, $L13$, $L12$, $L15$, $L16$, $L17$, $L18$, $L19$, $L20$ である．(1) を実行した結果を図 8.3 に示す．

定理 8.2 により，E と F は伝搬阻止ゲートである．(2-3) により，$L16$ と $L20$ に N が設定される．さらに，(2-1) により，$L17$, $L18$ にも N が設定される．したがって，この故障の診断点集合は $\{L10\}$ である．□

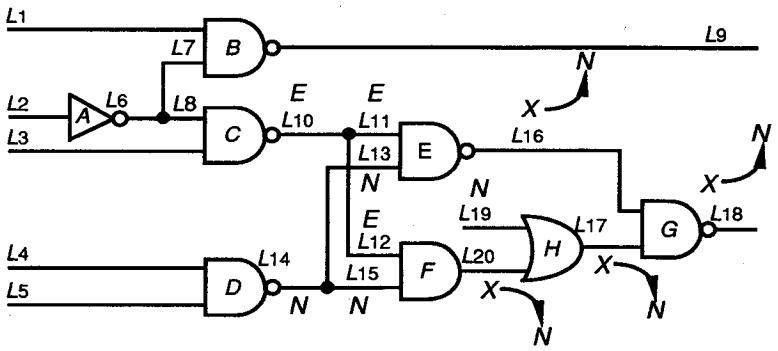


図 8.3　信号値の設定

8.1.3　故障診断手法

故障診断表の使い方は唯一ではない．まず，故障診断表を単なる故障辞書として使う場合について考える．例えば，表 8.1 に示す故障診断表を用いて，$6\times7\times C=42\times C$ の観測時間で故障診断を行なうことができる．この観測時間は電子ビームテスタで内部信号線を観測する時間を指す．なお，C は 1 本の内部信号線を電子ビームテスタで観測するために必要な時間である．

一般に，内部観測点数を m，テスト系列の長さを t とすると，観測に必要な時間は $m\times t\times C$ となる．今までの診断手法では，テスト系列の長さは一般に長いので，観測時間も長い．実用的な k-UCP 回路においては，k の値が 2 か 3 で t の値が 7 か 13 であるので，一般の故障辞書を利用した手法より観測時間が短い．したがって，故障診断表を単なる故障辞書として使っても，相当効率的である．

以下では，故障診断表に基づいて故障位置を特定するための観測点を決める新しい手法について述べる．

ここでいう故障診断とは，回路に存在可能な故障集合 F が与えられたとき，回路内のゲートの出力値を観測することにより，故障の有無の判定と故障が存在するときに故障位置を決定することである．ここでは，F 内のすべての故障の診断点集合の和集合を S，S 内の信号線 p を診断点とする故障の集合を $F(p)$，$F(p)$ 内の誤りビット位置が $b1, b2, \ldots, bt$ である故障の集合を $F(p(b1, b2, \ldots, bt))$ で表す．

ここで提案する故障診断手法の基本的な考え方は，S の中の 1 本の信号線 p を選んで観測し，そこに現われる系列の状態により，故障の候補となる F を小さくしていくことである．S 内のどの信号線を選んで観測を行なっていくかにより，故障位置指摘に必要な観測回数が異なってくる．したがって，観測点の選び方は非常に重要である．

F が誤りビット位置の異なる故障を含むとき，異常系列の有無と異常系列の誤りビット位

置により，Fをいくつかの部分集合に分けることができる．一方，Fが誤りビット位置の同じ故障のみを含むとき，異常系列の有無により，Fは2個の部分集合にしか分けられない．

ここで提案する故障診断手法では，F をなるべく多くの部分集合に分けることを優先にし，分けられる部分集合数が同じであるときには，各部分集合の大きさがなるべく同じになるような信号線を選ぶ方法を用いる．すなわち，観測点の選択にあたって，次のヒューリスティックな規則を用いる．

規則 1　S 内のすべての信号線 p について，$F(p)$ の故障を，故障の誤りビット位置により部分集合に分け，得られる部分集合の数が最大になるような信号線 p を観測点に選ぶ．

規則 2　S 内のすべての信号線 p について，$F(p)$ と F - F(p) の故障数の差の絶対値が最小になるような信号線 p を観測点に選ぶ．

提案する故障診断手法は次の手続きにより与えられる．

手続き FD（故障診断を行なう）

(1)　回路に存在可能なすべての故障の集合を F とし，診断点集合 S を求める．

(2)　S から規則 1 により観測点 p を選ぶ．

(3)　p を観測する．観測点に現われた系列が基本系列であれば，(4) へ移る．そうでなければ，(5) へ移る．

(4)　F - $F(p)$ を新しい F とする．この F が空であれば，回路が正常であるとして，終了する．そうでなければ，新しい F に対して S を求め，(5) へ移る．

(5)　p に現われた異常系列の誤りビットの位置が $b1, b2, \ldots, bt$ であるとき，$F(p(b1, b2, \ldots, bt))$ を新しい F とする．この F が 1 個の故障のみを含めば，故障が発見されたとして終了する．そうでなければ，新しい F に対して S を求め，(6) へ移る．

(6)　S から規則 2 により観測点を選ぶ．

(7)　p を観測する．観測点に現われた系列が基本系列であれば，F - $F(p)$ を新しい F とする．そうでなければ，$F(p)$ を新しい F とする．新しい F が 1 個の故障のみを含めば，故障が発見されたとして終了する．そうでなければ，新しい F に対して S を求め，(6) に戻る．　□

次の例は手続き FD の実行例を示す．

例 8.3　図 4.1 に示した回路に故障があるとし，表 8.1 に示した故障診断表を用いて故障位置の特定を行なう．故障位置特定の全過程を表 8.2 に示す．この表では，C.O.N. は現在

の観測点の番号を，O. P. は観測点を，O. R. は観測結果を，P. F. は可能な故障を，N. O. N. は次の観測点の番号を表す．また，観測結果として誤りビット位置が与えられる．基本系列が観測されたとき，観測結果として *NO* が記入される．

まず，$L18$ を観測する．もし誤りビット位置が6である異常系列が観測されたら，故障は *F-2* である．もし誤りビット位置が5と7である異常系列が観測されたら，故障は *A-2* または *E-3* である．この場合に，2番目の観測点として，$L20$ を観測する．これは N. O. N. により指示される．もし $L20$ で異常系列が観測されたら，故障は *A-2* で，そうでなければ，故障は *E-3* である．□

表 8.2　故障診断の過程

C. O. N.	O. R.	O. R.	P. F.	N. O. N.
1	$L18$	5, 7	*A-2, E-3*	2-1
		2, 4	*C-3, G-3*	2-2
		5	*C-2, G-2*	2-3
		3, 6	*D-3, H-3*	2-4
		2	*E-1, F-1*	2-5
		6	*F-2*	
		3	*G-1*	
		4	*H-1*	
		NO	*A-1, B-1, B-2, B-3, C-1, D-1, D-2, E-2, F-3, H-2*	2-6
2-1	$L20$	5, 7	*A-2*	
		NO	*E-3*	
2-2	$L20$	2.4	*C-3*	
		NO	*G-3*	
2-3	$L20$	5	*C-2*	
		NO	*G-2*	
2-4	$L20$	3, 6	*D-3*	
		NO	*H-3*	
2-5	$L20$	2	*F-1*	
		NO	*E-1*	
2-6	$L9$	6	*A-1*	
		4	*B-1*	
		7	*B-2*	
		3, 6	*B-3*	
		NO	*C-1, D-1, D-2, E-2, F-3, H-2*	3
3	$L10$	3	*C-1*	
		NO	*D-1, D-2, E-2, F-3, H-2*	4
4	$L14$	3	*D-1*	
		7	*D-2*	
		NO	*E-2, F-3, H-2*	5
5	$L16$	6	*E-2*	
		NO	*F-3, H-2*	6
6	$L17$	7	*H-2*	
		NO	*F-3*	

この例での平均観測時間は $7\times1.1\times C = 7.7\times C$ であり，単なる故障辞書として使用するときの $42\times C$ より少ない．

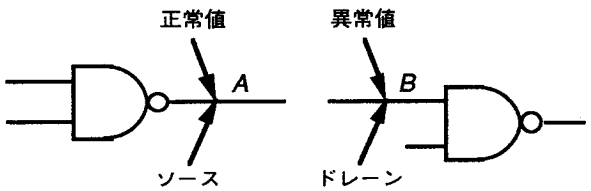
8.2　ガイディド・プローブ法による故障診断の効率化

ガイディド・プローブ法は電子ビームテスタを利用した故障診断システムで多用されている．この方法では，内部信号線を含む信号線の観測を行ないながら故障箇所を指摘するので，効率化をはかるため観測点を如何に少なくすることが重要である．以下では，ゲートの故障確率の概念を導入し，観測点の効率的な決定方法について提案する [61].

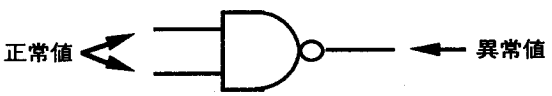
8.2.1　基本概念

8.2.1.1　ガイディド・プローブ法の原理

ガイディド・プローブ法では，内部信号線を含む信号線を観測しながら故障箇所にたどりつき故障診断を行なう．その基本は，最初に異常値が観測された信号線からバックトレースすることである．ガイディド・プローブ法により，図8.4に示すような断線や故障ゲートは特定できる [19].



(a)　断線



(b)　故障ゲート

図 8.4　カイディド・プローブ法の原理

ガイディド・プローブ法においては，1本の信号線がドレーンとソースという両端をもつと考える．図 8.4 (a) に示したように，ゲートに近い端をソース，もう一方の端をドレーンと呼ぶ．以下では，説明を分かりやすくするため，外部入力線には断線故障が生じないと仮定する．しかし，得られる結論は外部入力線に断線が生じる場合にも適用できる．

例 8.4　図8.5に示す回路の外部出力線に異常値があるとして，故障箇所を特定する．まず，$L1$ のドレーンを観測する．$L1$ のドレーンに異常値があれば，次に $L3$ のドレーンを観測する．$L3$ のドレーンに正常値があるとき，ゲート B が故障しているか，または $L1$ が断線しているかのいずれかである．このとき，$L1$ のソースを観測するとよい．$L1$ のソースに正常値があれば $L1$ が断線していることが分かり，$L1$ のソースに異常値があれば B が故障ゲートであることが分かる．□

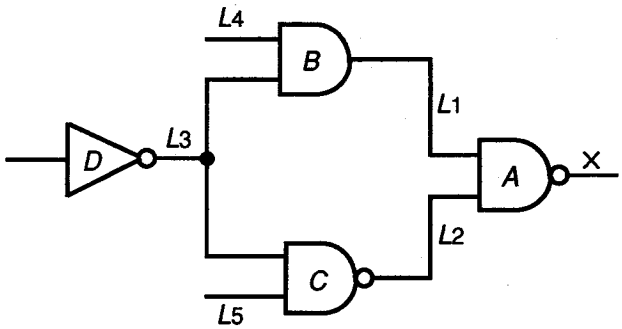


図 8.5　回路例

この例で示したように，ガイディド・プローブ法においては，いつも信号線のドレーンを観測すればよい．その結果は，すべての入力線のドレーンに正常値があり，出力線のドレーンに異常値があるゲートを発見することである．この場合，そのゲートの出力線のソースを観測すれば，そのゲートが故障しているかまたはその出力線が断線しているかを区別できる．したがって，ガイディド・プローブ法においては，まずゲート故障のみを仮定すればよい．発見した故障ゲートの出力線のソースをさらに観測すれば，完全な故障診断となる．以下では，ゲート故障のみを考えることにする．また，信号線を観測するとは，その信号線のドレーンを観測することを意味する．

8.2.1.2　被疑部分回路

ガイディド・プローブ法における信号線の観測は，通常電子ビームテスタにより行なわれるため，信号線の観測には相当な時間がかかり，観測点の数を極力少なくする必要がある．観測点の数は，回路構造，回路規模，および観測点の決定方法に依存する．

一般に，故障診断では，回路全体を対象とする必要はなく，故障が存在していると思われる部分回路，いわゆる被疑部分回路を対象にすれば十分である．ガイディド・プローブ法による故障診断を効率的に行なうためにまずすべきことは，被疑部分回路をなるべく小さくすることである．これを目的としたいくつかの方法が提案されている [65, 70] が，縮退故障に対してのみ有効なものが多く，ここで扱うゲート故障という一般的な故障モデルには適用で

きない．以下では，被疑部分回路を絞り込む一般的な方法について述べる．

組合せ回路の場合，回路全体を被疑部分回路とする必要はない．1本の外部出力線にのみ異常値があるとき，それのコーンを被疑部分回路とすればよい．また，1本以上の外部出力線に異常値があるとき，それらのコーンの共通部分を被疑部分回路とすればよい．信号線 L のコーンとは，L に到達できるすべての信号線とゲートからなる部分回路のことである．明らかに，組合せ回路の被疑部分回路は単一出力の組合せ回路であり，その外部出力線に異常値がある．

テストベクトル $T1, T2, \ldots, Tr$ を図8.6に示す一般の順序回路に印加したとき，外部出力線の Li に Tr で異常値があるとする．このような順序回路に対するガイディド・プローブ法による故障診断は，次の手続きにより行なうことができる．

手続き FDS（順序回路の故障診断を行なう）

(1)　$T1, T2, \ldots, Tr$ を印加しながらクロック信号線 *CLK* を観測する．すべてのテストベクトルに対して *CLK* に正常値があれば，Li を *LINE* とする．少なくとも1個のテストベクトルに対して *CLK* に異常値があれば，クロック信号線 *CLK* が故障していると分かり，終了する．

(2)　*LINE* のコーンを被疑部分回路としてガイディド・プローブ法による故障診断を行なう．その結果は，そのコーンに故障ゲートを発見したら，終了する．そうでなければ，あるフリップ・フロップの出力線に必ず異常値がある．そのフリップ・フロップを *FFi* とする．

(3)　$T1, T2, \ldots, Tr$ を印加しながら *FFi* の出力線を観測する．*FFi* の出力線がすべてのテストベクトルに対して正常値を出力すれば，フリップ・フロップの *FFi* が故障していると分かり，終了する．*FFi* の出力線がテストベクトル Tp $(p \leq r)$ に対して異常値を出力すれば，*FFi* の入力線を *LINE* とし，p を r とし，(2) に戻る．　□

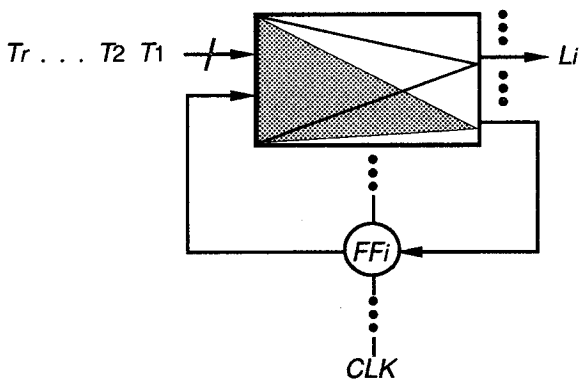


図8.6　カイディド・プローブ法による順序回路の故障診断

明らかに，観測を行なうとき，観測点を選択する必要のない場合もある．例えば，あるフリップ・フロップの出力線に異常値があれば，次に必ずそのフリップ・フロップの入力線を観測する．しかし，組合せ部分において，ある外部出力線のコーンに対して故障診断を行なうとき，複数の候補の中から次の観測点を選択する必要がある．つまり，被疑部分回路が外部出力線に異常値がある単一出力の組合せ回路であるときにのみ，観測点の決定方法を効率化する必要がある．

8.2.1.3　電子ビームテスタに基づくガイディド・プローブ法

最近では，ガイディド・プローブ法による故障診断システムでは，電子ビームテスタが一般的に利用されている[28]．電子ビームテスタにより，回路の最上位層にある信号線の状態を直接に観測できるが，下位層にある信号線を観測するとき，FIB (Focused Ion Beam) を用いてその信号線に通じる穴をあけてから観測する必要がある．

図8.7は電子ビームテスタに基づくガイディド・プローブ法による故障診断システムの一般構成を示す．回路内素子の接続データと前回の観測結果により，次の観測点が決定される．その信号線が最上位層にあれば観測を直接に行なうが，下位層にあればそれを露出させてから観測を行なう．さらに，観測値と期待値が比較され，一致するか否かなどの情報が次の観測点を決定するために利用される．

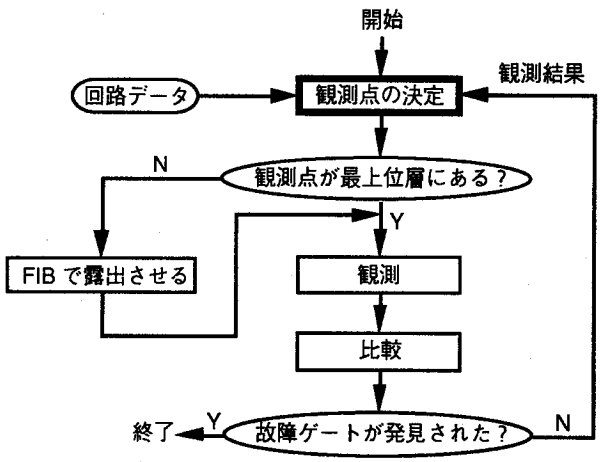


図8.7　電子ビームテスタに基づくガイディド・プローブ法による故障診断

このような故障診断システムでは，次の観測点を決定する方法は効率的でなければならない．つまり，回路全体の故障診断に必要な観測回数が少ない観測点の決定方法を使用しなければならない．

もっとも基本的な観測点の決定方法では，現在の被疑ゲートの1本の入力線を選んで観測する．例えば，図8.5の回路では，ゲート A の出力線に異常値があるとする．この場合，

現在の被疑ゲートがゲート A なので，まず $L1$ を観測する．もし $L1$ に異常値があれば，現在の被疑ゲートはゲート B になる．この場合，$L3$ を観測する．このようにして，故障ゲートを特定できる．以下では，この方法を EP 決定法と呼ぶ．

一般に，EP 決定法は多くの観測回数を必要とする．この方法を改良するため，いくつかのヒュリスティックが提案されているが，いずれにも大きな効果が期待できない [19].

8.2.2　観測点の効率的な決定方法

本節では，電子ビームテスタに基づくガイディド・プローブ法による故障診断システムについて調べ，観測点の効率的な決定方法が満たすべき条件を明確にした上で，観測点の効率的な決定方法を提案する．

8.2.2.1　基本条件

図8.7より明らかなように，観測点が決定され観測が始まると，次の観測点の決定作業に入ることが可能である．つまり，観測と決定を同時に行なうことが考えられる．しかし，この場合，今の観測結果が分からないので，観測値が期待値と等しい場合の次の観測点および観測値が期待値と等しくない場合の次の観測点の両方を決定しておく必要がある．次の観測点を決定しておくと，観測点の決定を待つ時間が省かれ，故障診断システム全体の効率が向上する．

1 個の観測点を決める時間を $t1$，FIB で穴をあける時間を $t2$，信号線を観測する時間を $t3$，観測値と期待値を比較する時間を $t4$ とする．観測点の決定を待つ時間が 0 になるため，次の条件を満たさなければならない．

条件 1（観測中の信号線が最上位層にあるとき）　$t1 < (t3 + t4) / 2$
条件 2（観測中の信号線が下位層にあるとき）　$t1 < (t2 + t3 + t4) / 2$

以下で明らかにするように，観測点の決定方法がよいほど実行時間も長くなる．このことから，電子ビームテスタに基づくガイディド・プローブ法による 2 段階の故障診断システムの一般構成を提案する．これを図 8.8 に示す．第 1 段階では，最上位層にある信号線のみを観測点の候補として考える．ここでの観測点の決定方法は，$t1 < (t3 + t4) / 2$ という条件を満たさなければならない．この段階での最終結果は 1 個の故障ゲートまたは可能な故障ゲートの集合である．後者の場合，第 2 段階の故障診断が必要である．この段階での観測点の決定方法は，$t1 < (t2 + t3 + t4) / 2$ という条件を満たさなければならない．

明らかに，第 1 段階と第 2 段階では，異なる決定方法を用いればよい．第 1 段階では，$(t3 + t4) / 2$ が約 5 秒 [66, 68] であるため，実行時間に対する制約が強い．第 2 段階では，$t2$ が約数分から数十分であるため，実行時間に対する制約が強くないが，FIB で穴をあけるコ

ストが高いので，観測回数を最小に抑えることが重要である．

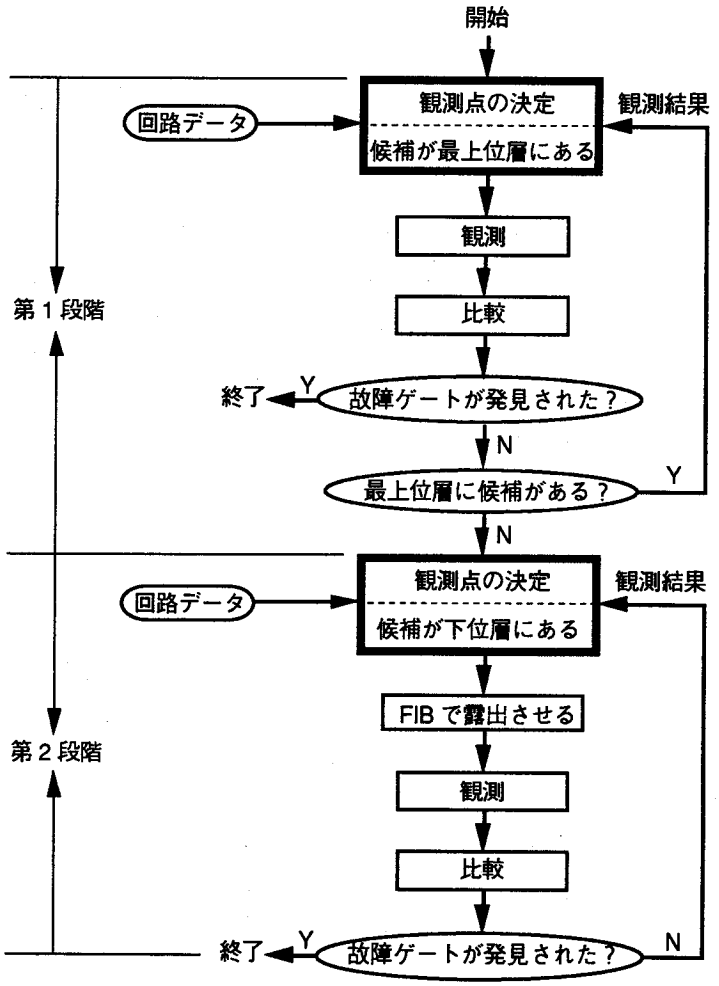
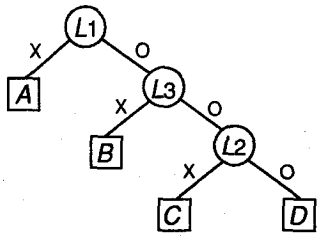


図 8.8　2段階の故障診断システム

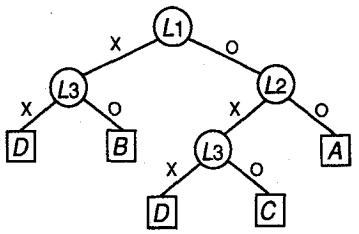
以下では，このような条件を満たす決定方法を提案する．その前に，観測木と故障確率の概念について述べる．

8.2.2.2　観測木と故障確率

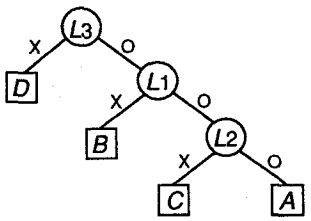
観測点の決定方法の効率は，1個の故障ゲートの位置を特定するために必要な平均観測回数で評価できる．故障診断の全過程を特殊な2分木である観測木で表すと，その平均観測回数を簡単に計算することができる．図8.5に示した回路のEP決定法による故障診断を表す観測木を図8.9 (b) に示す．観測木では，円は観測点を，正方形は故障ゲートを表す．また，異常値が観測されたとき（X）は左へ，正常値が観測されたとき（O）は右へ進む．



(a)　観測木でない2分木



(b)　EP 決定法の観測木



(c)　最適観測木

図 8.9　2分木

一般に，n 個のゲートからなる組合せ回路の場合，$(n-1)!$ 個の観測木がある．しかし，任意の2分木は必ずしも観測木ではない．例えば，図 8.5 に示した回路に対して図 8.9 (a) の2分木は観測木ではない．なぜなら，$L1$ に異常値があることは必ずしもゲート A が故障していることを意味しないからである．

さらに，回路内のすべてのゲートに同じ確率で故障が生じることは極めて稀である．それぞれのゲートが異なる故障確率をもつとすることが妥当である．この場合，ゲートの故障確率の決め方は観測点の決定方法の評価に影響を与える．次はその一例である．

例 8.5　図 8.9 (b) と (c) に示した2分木は図 8.5 に示した回路の観測木である．図 8.9 (b) の観測木に対応する決定方法での平均観測回数は $P = 2*PA + 2*PB + 3*PC + 5*PD$ であり，

図 8.9 (c) に対応する決定方法での平均観測回数は $Q = 3*PA + 2*PB + 3*PC + PD$ である．$PA = 2/8$, $PB = 3/8$, $PC = 2/8$, $PD = 1/8$ の場合，$P = 21/8$, $Q = 19/8$ である．この場合，図 8.9 (c) に示した観測木に対応する決定方法の方がよい．$PA = 6/12$, $PB = 3/12$, $PC = 2/12$, $PD = 1/12$ の場合，$P = 29/12$, $Q = 31/12$ である．この場合，図 8.9 (b) に示した観測木に対応する決定方法の方がよい．□

この例は故障確率の重要性を示している．各ゲートの故障確率を正確に決めることは，物理欠陥に関する統計的な研究が必要であるため，極めて困難である．本論文では，面積の大きいゲートがより高い確率で故障することに着目し，故障確率を決める近似的な方法を提案する．面積をトランジスタの数で評価すると，次の式が得られる．

ゲート G の故障確率 = ゲート G 内のトランジスタ数 / 被疑部分内のトランジスタ数

例えば，図 8.5 に示した回路は CMOS のゲート [1] から構成されているとすると，ゲート A, B, C, D のトランジスタ数はそれぞれ 4, 6, 4, 2 である．したがって，各ゲートの故障確率は $PA = 4/16 = 2/8$, $PB = 6/16 = 3/8$, $PC = 4/16 = 2/8$, $PD = 2/16 = 1/8$ である．

8.2.2.3　平衡決定法

本節では，第 1 段階で使用できる観測点の決定方法について述べる．第 1 段階では，最上位層にある信号線のみを候補信号線とする．

$G = \{Gi \mid i = 0, 1, 2, \ldots, n\}$ を被疑部分回路内のゲートの集合とする．ここでいう被疑部分回路は，外部出力線に異常値がある単一出力の組合せ回路である．$G0$ の出力線を被疑部分回路の外部出力線とする．$L = \{Li : Gi$ の出力線 $\mid i = 1, 2, \ldots, n\}$ とする．また，PGi を Gi の故障確率とし，$P(G) = PG0 + PG1 + PG2 + \ldots + PGn$ とする．

一般に，ある信号線を選択し観測したとき，その観測結果により，もとの可能な故障ゲートの集合をより小さい可能な故障ゲートの集合に縮小できる．観測点の効率的な決定方法は，毎回の観測結果により可能な故障ゲートの集合を大幅に縮小できるものと考えられる．

Li $(i = 1, 2, \ldots, n)$ を観測した結果により，G が G' または G'' に縮小される．G' は Li に異常値があるときの可能な故障ゲートの集合であり，G'' は Li に正常値があるときの可能な故障ゲートの集合である．Li の観測結果が分かるまで，故障ゲートが G' にあるか G'' にあるかが分からない．ここで，G' が G'' より大きいとする．この場合，もし Li に異常値があれば，可能な故障ゲートの集合が G' となる．G' が大きく G'' が小さいので，Li を観測することは得策ではないことになる．つまり，G' と G'' の大きさがなるべく等しくなるような Li を観測した方がよい．

さらに，ゲートの故障確率が用いられる場合，可能な故障ゲートの集合の単純な大きさで

はなく，確率付きの大きさを用いる必要がある．つまり，Gの大きさを$P(G)$で表すことは望ましい．したがって，$P(G)$と$P(G')$がなるべく等しくなるようなLiを観測線に選んだ方がよい．これは提案する観測点の平衡決定法の基本的な考え方である．

平衡決定法では，故障診断表と呼ばれる表が用いられる．図8.10に示す回路の故障診断表を表8.3に示す．このような表は故障シミュレーションにより求められる．

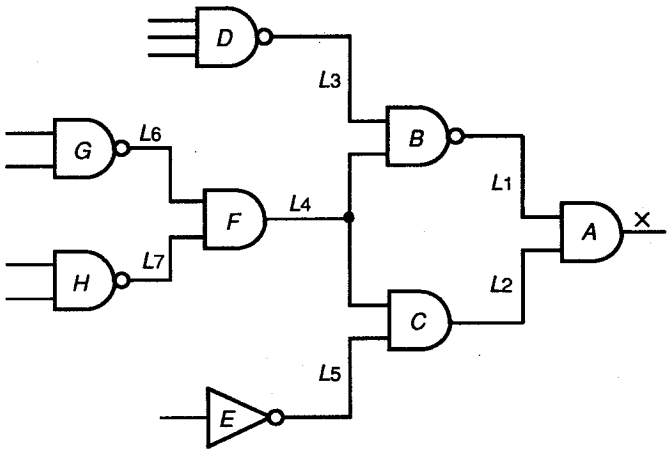


図8.10　回路例

表8.3　故障診断表

P	N	G \ L	L1	L2	L3	L4	L5
3/19	1	A	0	0	0	0	0
2/19	2	B	1	0	0	0	0
3/19	3	C	0	1	0	0	0
3/19	4	D	1	0	1	0	0
1/19	5	E	0	1	0	0	1
7/19	6	F, G, H	1	1	0	1	0

故障診断の第1段階では，最上位層にある信号線のみを観測点の候補とするため，複数のゲートは同様な故障動作をする可能性がある．例えば，図8.10の回路において，$L1, L2, L3, L4, L5$が最上位層にあり，$L6$と$L7$が下位層にあるとする．この場合，最上位層にある信号線から見ると，ゲートF, G, Hは同様な故障動作をする．故障診断表では，これらのゲートを一つのグループにまとめる．表8.3に示したように，列qと行rの交差点にある1は，グループrにあるゲートが故障したとき信号線qに異常値が現われることを意味し，列qと行rの交差点にある0は，グループrにあるゲートが故障しても信号線qに異常値が現われないことを意味する．列Pには各グループの故障確率が示されている．また，列N

とGにはそれぞれグループの番号と各グループの内容が示されている．

平衡決定法は，故障診断表が用いられる．例えば，$L1$ について，$G'=\{B, D,$ グループ5$\}$ と $G''=\{A, C, E\}$ となる．$P(G')$ は $PB+PD+PF+PG+PH=2/19+3/19+3/19+2/19+2/19=12/19$ であり，$P(G'')$ は $PA+PC+PE=3/19+3/19+1/19=7/19$ である．ここで，$P(G')$ と $P(G'')$ の差の絶対値を $dp(L1)$ で表すと，$dp(L1)=5/19$ となる．同様にして，$dp(L2)=3/19$, $dp(L3)=13/19$, $dp(L4)=5/19$, $dp(L5)=17/19$ が得られる．したがって，平衡決定法では，$L2$ を観測線に選ぶ．

$L2$ で異常値が観測されたとすれば，可能な故障ゲートの集合は $\{B, D,$ グループ5$\}$ となる．これにより，もとの故障診断表を表8.4に示す表に縮小できる．この場合，被疑部分回路が小さくなったので，ゲートの故障確率を計算しなおす必要がある．被疑部分回路がゲート C, E, F, G, H からなるので，新しい故障確率は $PC=3/11$, $PE=1/11$, $PF=3/11$, $PG=2/11$, $PH=2/11$ である．この縮小された故障診断表を利用してさらに観測点を選んでいくことにより故障診断を行なうことができる．

表8.3　縮小された故障診断表

P	N	G ＼ L	$L1$	$L5$
3/11	1	C	0	0
1/11	2	E	0	1
7/11	3	F, G, H	1	0

平衡決定法は次の手続きで表すことができる．

手続き BP（平衡法で観測点を決める）

(1)　もとの被疑部分回路に対して故障診断表を作り，それを TABLE とする．

(2)　TABLE 内の信号線 Li に対して，$dp(Li)$ を計算する．$dp(LINE)$ が最小である $LINE$ を観測点に選ぶ．

(3)　$LINE$ を観測する．

(4)　観測結果により，被疑部分回路を小さくし，故障確率を再計算し，故障診断表を縮小する．縮小された故障診断表を TABLE とする．

(5)　TABLE が1個の可能な故障ゲートのグループしかもたないなら，終了する．そうでなければ，(2)へ戻る．　□

この観測点の決定方法は第1段階で使用される．この段階では，最上位層にある信号線のみを観測点の候補とするので，その診断結果は故障ゲートではなく可能な故障ゲートの集合である可能性がある．このようにして得られた可能な故障ゲートの集合の中からさらに故

障ゲートを特定するため，第2段階で下位層にある信号線をFIBで露出させ観測する必要がある．FIBでの露出作業はコストが高く所要時間が長いので，第2段階で用いられる観測点の決定方法はなるべく小さい平均観測回数を保証しなければならない．

8.2.2.4　最適決定法

本節では，平均観測回数を最小にするような観測点の決定方法，いわゆる最適決定法について提案する．

明らかに，観測木において，ある葉から根までの距離はその葉に対応する故障ゲートを特定するための観測回数である．観測木より，それに対応する決定方法における平均観測回数を計算することが簡単である．

通常，同じ回路に対しても異なる決定方法における平均観測回数は異なる．平均観測回数が最小である観測点の決定方法を最適決定法と呼ぶ．最適決定法に対応する観測木を最適観測木と呼ぶ．

次に最適観測木を求める方法について述べる．この方法では，最小平均観測回数を直接に求めるが，その過程より最適観測木を求めることができる．

$G = \{Gi \mid i = 0, 1, 2, \ldots, n\}$ を被疑部分回路内のゲートの集合とする．ここでいう被疑部分回路は，外部出力線に異常値がある単一出力の組合せ回路である．$G0$ の出力線を被疑部分回路の外部出力線とする．$L = \{Li : Gi$ の出力線$\mid i = 1, 2, \ldots, n\}$ とする．また，PGi を Gi の故障確率とし，$P(G) = PG0 + PG1 + PG2 + \ldots + PGn$ とする．この状態での最小平均観測回数を $f(L/G)$ で表す．

L 内の各信号線は最初の観測点の候補である．Li $(i = 1, 2, \ldots, n)$ を最初の観測点にした場合について考える．Li に異常値があれば，可能な故障ゲートの集合は $G' = \{Gi$ と Gi へ到達できるゲート$\}$ となり，観測点の候補集合は $L' = \{Li$ を除く G' 内のゲートの出力線$\}$ となる．Li に正常値があれば，可能な故障ゲートの集合は $G'' = \{Gi$ へ到達できないゲートとこれらのゲートへ到達できるゲート$\}$ となり，観測点の候補集合は $L' = \{G''$ 内のゲートの出力線$\}$ となる．つまり，L 内のある信号線を観測したことにより，もとの被疑部分回路が2個のより小さい被疑部分回路に縮小できる．これらの小さい被疑部分回路の最小平均観測回数が求まれば，もとの被疑部分回路の最小平均観測回数も次の式より求まれる．

$$f(L/G) = \min_{Li\,(i = 1, 2, \ldots, n)} \{P(G) + f(L'/G') + f(L''/G'')\}$$

L が空集合であるとき，$f(L/G) = f(-) = 0$ とする．これを初期条件として最小平均観測回を求めることができる．次にその例を示す．

例 8.6　図 8.5 に示した回路を被疑部分回路とすると，$G=\{A, B, C, D\}$, $L=\{L1, L2, L3\}$, $PA=2/8$, $PB=3/8$, $PC=2/8$, $PD=1/8$ である．

最小平均観測回数が $f(\{L1, L2, L3\}/\{A, B, C, D\})$ である．$L1$ に異常値がある場合，可能な故障ゲートは B と D である．つまり，$G'=\{B, D\}$, $L'=\{L3\}$ である．$L1$ に正常値がある場合，可能な故障ゲートは A, C, D である．つまり，$G''=\{A, C, D\}$, $L''=\{L2, L3\}$ である．同様に，$L2$ に異常値がある場合には，$G'=\{C, D\}$, $L'=\{L3\}$ であり，$L2$ に正常値がある場合には，$G''=\{A, B, D\}$, $L''=\{L1, L3\}$ である．また，$L3$ に異常値がある場合には，$G'=\{D\}$, $L'=\{-\}$ であり，$L3$ に正常値がある場合には，$G''=\{A, B, C\}$, $L''=\{L1, L2\}$ である．したがって，最小平均観測回数は $1+f(\{L3\}/\{\{B, D\})+f(\{L2, L3\}/\{A, C, D\})$, $1+f(\{L3\}/\{C, D\})+f(\{L1, L3\}/\{A, B, D\})$, $1+f(\{-\}/\{D\})+f(\{L1, L2\}/\{A, B, C\})$ の中で一番小さい値である．これを次のように表す．

$f(\{L1, L2, L3\}/\{A, B, C, D\})$
$L1$: $1+f(\{L3\}/\{\{B, D\})+f(\{L1, L3\}/\{A, C, D\})$
$L2$: $1+f(\{L3\}/\{C, D\})+f(\{L1, L3\}/\{A, B, D\})$
$L3$: $1+f(\{-\}/\{D\})+f(\{L1, L2\}/\{A, B, C\})$

このようにして，もとの回路の最小平均観測回数を求める問題をより小さい回路の最小平均観測回数を求める問題におきかえることができる．場合により，問題をさらに細分化することが必要である．この問題に関するすべての展開を次に示す．( ) 内の数字は展開の順序を，<> 内の数字は計算の順序を示す．

(1) $f(\{L1, L2, L3\}/\{A, B, C, D\})=19/8$　<17>
　$L1$: $1+f(\{L3\}/\{\{B, D\})+f(\{L1, L3\}/\{A, C, D\})=20/8$　<7>
　$L2$: $1+f(\{L3\}/\{C, D\})+f(\{L1, L3\}/\{A, B, D\})=21/8$　<12>
　$L3$: $1+f(\{-\}/\{D\})+f(\{L1, L2\}/\{A, B, C\})=19/8$　<16>
(2) $f(\{L3\}/\{B, D\})=4/8$　<1>
　$L3$: $4/8+f(\{-\}/\{D\})+f(\{-\}/\{B\})=4/8$　<1>
(3) $f(\{L2, L3\}/\{A, C, D\})=8/8$　<6>
　$L2$: $5/8+f(\{L3\}/\{C, D\})+f(\{-\}/\{A\})=8/8$　<3>
　$L3$: $5/8+f(\{-\}/\{D\})+f(\{L2\}/\{A, C\})=9/8$　<5>
(4) $f(\{L3\}/\{C, D\})=3/8$　<2>
　$L3$: $3/8+f(\{-\}/\{D\})+f(\{-\}/\{C\})=3/8$　<2>
(5) $f(\{L2\}/\{A, C\})=4/8$　<4>
　$L2$: $4/8+f(\{-\}/\{C\})+f(\{-\}/\{A\})=4/8$　<4>

(6) $f(\{L2, L3\}/\{A, B, D\}) = 10/8$　<11>

$L1$: $6/8 + f(\{L3\}/\{B, D\}) + f(\{-\}/\{A\}) = 10/8$　<13>

$L3$: $6/8 + f(\{-\}/\{D\}) + f(\{L1\}/\{A, B\}) = 11/8$　<14>

(7) $f(\{L1\}/\{A, B\}) = 5/8$　<9>

$L1$: $5/8 + f(\{-\}/\{B\}) + f(\{-\}/\{A\}) = 5/8$　<9>

(8) $f(\{L1, L2\}/\{A, B, C\}) = 11/8$　<15>

$L1$: $7/8 + f(\{-\}/\{B\}) + f(\{L2\}/\{A, C\}) = 11/8$　<13>

$L2$: $7/8 + f(\{-\}/\{C\}) + f(\{L1\}/\{A, B\}) = 12/8$　<14>

最小平均観測回数は 19/8 = 2.375 である．図 8.9 (c) に最適観測木が示されている．例 8.5 で示したように，同じ回路に対して EP 決定法で故障診断を行なうとき，平均観測回数は 21/8 = 2.625 である．

最適観測木は次のように構築できる．$f(\{L1, L2, L3\}/\{A, B, C, D\}) = 1 + f(\{-\}/\{D\}) + f(\{L1, L2\}/\{A, B, C\}) = 19/8$ であるので，最初の観測点は $L3$ である．またこの式から，$L3$ に異常値があれば故障ゲートが D であり，そうでなければ可能な故障ゲートの集合は $\{A, B, C\}$ であることが分かる．さらに，$f(\{L1, L2\}/\{A, B, C\}) = 7/8 + f(\{-\}/\{B\}) + f(\{L2\}/\{A, C\}) = 11/8$ であるので，次の観測点は $L1$ である．このようにして，図 8.9 (c) に示した最適観測木が得られる．□

8.2.3　実験結果と考察

ISCAS 1985 のベンチマーク組合せ回路 [71] に対して，EP 決定法，平衡決定法および最適決定法による故障診断を行ない，平均観測回数を求めた．使用した言語は C 言語であり，使用したコンピュータは富士通の S-4/LC（12.5 MIPS）である．

1 個のベンチマーク回路に対して，1 本ないし 2 本の外部出力線を選んでそのコーンに対して平衡決定法および EP 決定法による故障診断における平均観測回数を求めた．表 8.5 にそれぞれのコーンのトランジスタ数とゲート数を示す．この表において，BP は平衡木決定法を，EP は EP 決定法を表す．

実験結果より，平衡決定法は EP 決定法より平均観測回数が少ないことが分かる．2,000 ものゲートからなる被疑部分回路に対しても，10 回前後の観測で故障診断ができた．実際の故障診断では被疑部分回路を通常 1K ゲート程度に絞り込んでから電子ビームテスタで故障位置を特定するので，提案した平衡決定法は実用的な故障診断手法であることが分かる．また，平衡決定法により 1 個の観測点を決める平均時間が 5 秒以下なので，この方法は，電子ビームテスタに基づくガイディド・プローブ法による 2 段階の故障診断過程の第 1 段階に適用できる．

表 8.6 は最適決定法による故障診断の実験結果を示す．この方法は，最小の平均観測回数

が得られ，下位層にある信号線を観測しなければならないときに有用である．

表 8.5　実験結果 (1)

circuit	case	# of transistors	# of gates	average # of probed lines		time per probed line (sec.)	
				BP	EP	BP	EP
C432	# 1	832	146	9.16	14.95	0.035	0.001
	# 2	802	142	8.73	12.60	0.034	0.002
C499	# 1	1312	102	7.31	13.61	0.027	0.001
C880	# 1	632	130	7.53	20.98	0.037	0.002
	# 2	574	119	7.41	20.27	0.034	0.002
C1355	# 1	1316	322	8.75	23.03	0.131	0.003
C1908	# 1	2112	549	9.75	19.73	0.396	0.009
	# 2	1938	525	9.51	24.20	0.326	0.007
C2670	# 1	3874	828	10.09	20.97	0.789	0.006
	# 2	2538	503	9.41	20.00	0.310	0.006
C3540	# 1	6618	1458	11.12	27.75	1.814	0.007
	# 2	6498	1433	11.09	26.73	1.750	0.007
C5315	# 1	4102	937	10.09	28.34	0.875	0.009
	# 2	2170	494	9.18	21.90	0.283	0.011
C6288	# 1	9756	2327	10.82	56.56	4.651	0.008
	# 2	7072	1689	11.28	40.13	2.369	0.008
C7552	# 1	4764	1069	10.74	20.19	1.490	0.017
	# 2	3006	676	9.82	21.48	0.558	0.015

表 8.6　実験結果 (2)

circuit	# of gates	# of transistors	average # of probed lines			time per probed line (min.)
			BEST	EP	BP	BEST
C1	8	30	3.03	3.70	3.03	2.17
C2	15	62	3.91	4.76	4.56	3.16

8.3　まとめ

本章ではまず，CMOS の k-UCP 回路のスタック・オープン故障の診断手法について述べた．k-UCP 回路にテスト集合を印加したとき，故障ゲートの出力線および他のいくつかの信号線に基本系列と異なる異常系列が現われる．これらの異常系列の位置および異常系列が基本系列と異なるビットの位置などの情報により，回路内の任意の単一スタック・オープン故障の位置を指摘できることを示した．

提案した故障診断手法では，毎回1本の信号線を選んでそれを観測する．観測結果により可能な故障集合を小さくしていく．この操作は可能な故障集合の故障数が1個になるまで繰り返される．提案した故障診断手法の特徴は，故障診断表を利用して全観測回数を少なくすることである．故障診断表にはすべての故障に関する異常系列の位置および誤りビットの位置などの診断用情報が含まれる．故障診断表を用いることにより，可能な故障集合を一番小さくすることができる信号線を簡単に見つけることができる．毎回このような信号線を観測点にすれば全観測回数が少なくて済むものである．

また，故障診断表の効率的な生成法を提案した．伝搬阻止ゲートの必要十分条件を利用し，三つの記号を用いた故障シミュレーションにより異常系列の位置が求められる．これは，ビットごとに行なわれる従来の故障シミュレーション手法に比べ非常に効率的である．

さらに，一般の論理回路の故障診断にも適用できる電子ビームテスタに基づくガイディド・プローブ法による2段階の故障診断過程を提案した．第1段階では，最上位層にある信号線のみを観測点の候補とし，故障診断表を利用した高速な観測点決定手法を用いて被疑部分回路を効率よく絞り込む．第2段階では，下位層にある信号線をコストの高いFIBで露出させる必要があるので，最小の平均観測回数が得られる観測点決定手法を用いる．その結果，全体の故障診断を効率よく行なうことができる．故障診断の効率をさらに向上させるため，ゲートの故障確率の概念を導入した．実験では，2,000ものゲートを含む回路に対してもわずか10回前後の観測で故障位置を特定し，提案した手法の有効性を確かめた．

9 総括

本論文は，全可観測な環境でのテスト容易な論理回路の特徴，性質および構成法についての研究成果について述べたものである．

電子ビームプロービング技術を用いることにより，回路内の最上位層にある信号線を観測することができる．この技術を背景に，回路のすべてのゲートの出力線が観測できるとするテスト環境は全可観測な環境である．電子ビームプロービング技術を使用する前提条件として，被観測回路が短いテスト系列をもち，かつその系列を回路に繰り返し印加することができなければならない．このため，全可観測な環境においてもテスト容易化設計が必要である．

第4章では，全可観測な環境でのテスト容易な組合せ回路として，k-UCP回路およびk-R回路を提案した．k-UCP回路はNOTゲートとk入力の基本ゲートで構成され，かついくつかの構造上の特徴をもつ．そのため，k-UCP回路のすべての縮退故障とスタック・オープン故障は，全可観測な環境においてそれぞれ長さ$k+1$と$k(k+1)+1$のテスト系列で検出できる．実際のkの値が2または3なので，テスト系列は非常に短い．次に，k-UCP回路のすべての縮退故障を検出するには，一部のゲートの出力線を観測するだけでよいことを明らかにした．それは，k-UCP回路にテスト集合を印加したとき，一部の縮退故障の影響が外部出力線に伝わるからである．このよう故障を見つけることは，クロスチエック法などで回路内に観測点を設けるために有用である．さらに，k-UCP回路におけるゲートの入力線数がkであるという制限をなくすため，k-R回路を提案した．k-R回路はNOTゲート，および入力線数がk以下の基本ゲートで構成される．k-R回路のすべての縮退故障は，全可観測な環境において長さ$k+1$のテスト系列で検出できることを示した．

第5章では，任意の組合せ回路をk-UCP回路に変換する手法について述べた．この変換は3段階に分けて行なわれる．まず最初の段階で，与えられた回路をk-U回路に変換する．次の2段階で，k-U回路をk-UP回路とk-UC回路に変換する．この中でもっとも困難な変換はk-U回路からk-UC回路への変換である．この変換ではまず，k-U回路が$k+1$色解をもつかどうかを決定する必要がある．この決定問題はNP完全問題であるため，ヒューリスティックな手法でその解決を図った．基本的な考え方は，外部入力線へのすべて

の色の割り当ての中から彩色解をもたらすようなものを探索することである．この探索を速くするために，外部入力線のヒューリスティックな選び方，および色の含意操作などを提案した．回路変換の実験結果より，与えられた回路を k-UC-NAND または k-UC-NOR 回路に変換するためのオーバーヘッドは様々なゲートからなる一般の k-UCP 回路に変換するためのオーバーヘッドより少ないこと，および与えられた回路が k 入力 NAND または NOR ゲートで構成される場合，それを k-UC-NAND 回路または k-UC-NOR 回路に変換するためのオーバーヘッドが非常に小さいことが分かった．

第6章では，全可観測な環境でのテスト容易な順序回路である k-UCP 順序回路と k-UCP スキャン回路について述べた．それぞれの回路の組合せ部分は k-UCP 回路である．また，k-UCP 順序回路においては，そのフリップ・フロップの入力側の構造に制限が加えられており，k-UCP スキャン回路においては，そのスキャンパスの組み方が工夫されている．その結果，k-UCP 順序回路内のすべての縮退故障は $3(k+1)$ 個のテストベクトルでテストでき，k-UCP スキャン回路内のすべての縮退故障は $k+1$ 個のテストベクトルでテストできる．また，それらのテストベクトルを回路に連続的に繰り返し印加することができる．このため，提案した回路は電子ビームテスタを用いた故障診断に適している．

第7章では，任意の順序回路を k-UCP 順序回路に変換する手法，および k-UCP スキャン回路に変換する方法を提案した．これらの変換では，まず最初に順序回路の組合せ部分を k-UCP 回路に変換する．次に，順序回路のフリップ・フロップ周りに関する調整を行なう．k-UCP 順序回路に変換するとき，フリップ・フロップの入力線へ必要に応じて入力ゲートを挿入するだけでよい．k-UCP スキャン回路に変換するとき，フリップ・フロップのすべての出力線の極性が同じで，色の番号が k-有効系列を構成するように付加ゲートで調整しなければならない．本章では，それをなるべく少ない付加ゲートで行なう手法を提案した．実験結果より，もとの順序回路がスキャンパスをもち，かつその組合せ部分が k 入力 NAND または NOR ゲートで構成される場合，それを k-UCP スキャン回路に変換するためのオーバーヘッドが小さく実用範囲内であることが分かった．

第8章ではまず，CMOS の k-UCP 回路のスタック・オープン故障の診断手法について述べた．提案した故障診断手法では，毎回 1 本の信号線を選んでそれを観測する．観測結果により可能な故障集合を小さくしていく．この操作は可能な故障集合の故障数が 1 個になるまで繰り返される．提案した故障診断手法の特徴は，故障診断表を利用して全観測回数を少なくすることである．故障診断表にはすべての故障に関する異常系列の位置および誤りビットの位置などの診断用情報が含まれる．故障診断表を用いることにより，可能な故障集合を一番小さくすることができる信号線を簡単に見つけることができる．毎回このような信号線を観測点にすれば全観測回数が少なくて済むものである．本章ではまた，伝搬阻止ゲートの必要十分条件を利用し，三つの記号を用いた故障シミュレーションにより故障診断表の効率的な生成法を提案した．これは，ビットごとに行なわれる従来の故障シミュレーション

手法に比べ非常に効率的である．さらに，一般の論理回路の故障診断にも適用できる電子ビームテスタに基づくガイディド・プローブ法による2段階の故障診断過程を提案した．第1段階では，最上位層にある信号線のみを観測点の候補とし，故障診断表を利用した高速な観測点決定手法を用いて被疑部分回路を効率よく絞り込む．第2段階では，下位層にある信号線をコストの高いFIBで露出させる必要があるので，最小の平均観測回数が得られる観測点決定手法を用いる．その結果，全体の故障診断を効率よく行なうことができる．故障診断の効率をさらに向上させるため，ゲートの故障確率の概念を導入した．実験では，2,000ものゲートを含む回路に対してもわずか10回前後の観測で故障位置を特定し，提案した手法の有効性を確かめた．

今後の課題として，以下のものが挙げられる．

(1) 本論文で提案したテスト容易な論理回路に関する制約条件をさらに緩め，テスト系列長が少々長くなる可能性があるが，回路変換によるオーバーッヘドが少ないようなテスト容易な論理回路を提案する．

(2) 回路の故障診断のもっとも困難な部分のみをテスト容易な論理回路に変換することはオーバーッヘドが少なくより現実的であるといえる．そのため，回路全体ではなくその一部分が本研究で提案したテスト容易な論理回路である場合の回路全体のテスト手法を考案する．

(3) 全可観測な環境ではなく，通常のテスト環境でのテスト容易な論理回路について考案する．

集積回路は産業と社会のあらゆる分野でなくてはならない存在となっている．特に最近では，各種の目的に製造される集積回路いわゆる特定用途向け集積回路（ASIC）へのニーズが非常に強く，それを短期間・低費用で開発しなければならない．そこで，開発・製造に必要な故障診断を如何に迅速に行なうか，量産時の故障検査コストを如何に抑えるかが重要なポイントとなっている．本研究で提案したテスト容易な論理回路はいずれも短いテスト系列でテストを効率よく行なうことができるため，VLSIの短期間・低費用の開発と製造に貢献するものと期待できる．

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なったものである．本研究を遂行するにあたり終始御指導を賜り，有益な議論および御助言を頂き，また，５年半にわたる留学生活のあらゆる面においても終始大変お世話になりました樹下行三教授に心より感謝致します．

本論文の作成にあたって詳細な御検討，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豊田順一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，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岩崎裕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，板崎徳禎助手，梶原誠司助手，東京都立大学三浦幸也助手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．ここに記して感謝致します．

参考文献

[1] D. A. Pucknell and K. Eshraghian: *Basic VLSI Design: Systems and Circuits*, Second Edition, Prentic-Hall 1987.

[2] T. W . Williams and K. P. Parker: "Design for Testability - a Survey", *Proc. IEEE*, Vol. 71, No.1, pp. 311-325, 1983.

[3] S. C. Seth and V. D. Agrawal: "Cutting Chip-Testing Costs", *IEEE Spectrum*, pp. 38-45, April 1985.

[4] E. J. McCluskey and F. Buelow: "IC Quality and Test Transparency," *Proc. ITC 1988*, pp. 295-301, 1988.

[5] 樹下行三，浅田邦博，唐津修：*VLSI の設計 II*，岩波書店, 1985.

[6] 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.

[7] 玉本英夫：*論理回路の故障診断*，日刊工業新聞社, 1983.

[8] J. P. Roth: "Diagnosis of Autamata Failures: A Calculus and a Method", *IBM J. Res. Develop.*, pp. 278-291, July, 1966.

[9] P. Goal: "An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits", *IEEE Trans. Comput., Vol. C-30, No.3*, pp.215-222, Mar. 1981.

[10] H. Fujiwara and T. Shimono: "On the Acceleration of Test Generation Algorithms", *IEEE Trans. Comput.*, pp.1137-1144, Dec. 1983.

[11] Y. Takamatu and K. Kinoshita: "CONT: A Concurrent Test Generation Algorithm", *Proc. FTCS-16*, pp.22-27, 1986.

[12] M. H. Schulz and E. Auth: "Advanced Automatic Test Pattern Generation and Redundancy Identification Techniques", *Proc. FTCS-18*, pp.30-35, 1988.

[13] V. D. Agrawal and S. C. Seth: *Test Generation for VLSI Chips*, IEEE Computer Society Press, 1988.

[14] 樹下行三："VLSI のテスト容易化設計技術の研究動向"，*情報処理学会学会誌，Vol. 30, No. 12*, pp. 1451-1460, 1989.

[15] E. J. McCluskey: *Logic Design Principles with Emphasis on Testable Semicustom Circuits*, Prentice-Hall, Inc., 1986.

[16] H. Fujiwara: *Logic Testing and Design for Testability*, The MIT Press, 1985.

[17] P. H. Bardell, W. H. McAnney, and J. Savir: *Built-In Test for VLSI*, Wiley-Intersince, 1987.

[18] R. G. Bennetts: *Design of Testable Logic Circuits*, Addison-Wesley publishing, Co., 1984.

[19] M. Abramovici, M. A. Breuer, and A. D. Friedman: *Digital Testing and Testable Design*, Computer Science Press, 1990.

[20] P. K. Lala: *Fault Tolerant & Fault Testable Hardware Design*, Prentice-Hall International, Inc., London, 1985.

[21] T. Gheewala: "CrossCheck: A Cell Based VLSI Testability Solution", *Proc. DAC-26*, pp. 706-709, June 1989.

[22] S. J. Chandra, T. Ferry, T. Gheewara and K. Pierce: "ATPG Based on a Novel Grid-Address-

able Latch Element", *Proc. DAC-28*, pp. 282-286, June 1991.

[23] 古川泰男，稲垣雄史："LSIの非接触診断技術"，*電学論 C, Vol. 107, No. 3*, pp. 245-250.

[24] E. Walfgang: "Electron Beam Testing", *Handbook of Advanced Semiconductor Technology and Computer Systems*, Van Norstand Reinhold Co., 1987.

[25] 裏克己，藤岡宏："電子ビームプロービング"，*電学論，102*, pp. 29-35.

[26] N. K. Jha and S. Kundu: *Testing and Reliable Design of CMOS Circuits*, Kluwer Academic Publishers, 1990.

[27] G. S. Plows and W. C. Nixon: "Strobosopic Scanning Eletron Microscopy", *ibid, Vol. 2, No. 1*, p. 595, 1968.

[28] E. Wolfgang, R. Linder, P. Fazekas, and H. Feuerbaum: "Electron-Beam Testing of VLSI Circuits", *IEEE J. Solid-States Circuits, Vol. SC-14, No. 2*, pp. 471-481, April 1979.

[29] 下野武志，渡辺純子，木村敬，加藤俊弘，村瀬真道："EBテスタによるLSIの故障解析のためのテストパターンの自動生成手法"，*日本学術振興会第132委員会第113回研究回資料集*，pp. 52-56, 1990.

[30] A. Tamata and N. kuji: "Integrating an Electron-Beam System into VLSI Fault Diagnosis", *IEEE Design and Test of Computers*, pp.23-29, Aug. 1986.

[31] 山口昇，佐藤司，戸所秀男，萩原吉宗，坂本隆："電子ビームテスタを用いたVLSIの故障探索法の基礎検討"，*信学技報，Vol. 89, No. 71*, pp. 9-16, 1989.

[32] 寺元光生："電子ビームテスタ用テストパターン生成法"，*信学技報，Vol. 88, No. 42*, pp. 45-50, 1988.

[33] T. Yano, H. Okamoto: "Fast Fault Diagnostic Method Using Fault Dictionary For Electron Beam Tester", *Proc. ITC 1987*, pp. 561-565, Sep. 1987.

[34] J. P. Hayes: "On Modifying Logic Networks to Improve Their Diagnosability", *IEEE Trans. Comput., Vol. C-23, No. 1*, pp. 56-62, Jan. 1974.

[35] K. K. Saluja and S. M. Reddy: "On Minimally Testable Logic Networks", *IEEE Trans. Comput., Vol. C-23, No. 5*, pp. 552-554, May 1974.

[36] 樹下行三，温暁青，スダーカM．レディ："全可観測な環境でのNAND回路の故障診断"，*電子情報通信学会論文誌 DI, Vol. J73, No. 2*, pp. 245-252, 1990.

[37] K. Kinoshita, W. Xiaoqing and S.M. Reddy: "Diagnostic Testing of NAND Circuits under Totally Observable Condition", *Proc. JFTCS'89*, pp. 69-74, Chongqing, China, June 1989.

[38] W. Xiaoqing and K. Kinoshita: "Testable Design of Logic Circuits under Highly Observable Condition," *IEEE Trans. Comput.*, Vol. 41, No. 5, pp. 654-659, May 1992.

[39] W. Xiaoqing and K. Kinoshita: "Fault Detection and Diagnosis of *k* - UCP circuits under Totally Observable Condition", *Proc. FTCS-20*, pp. 382-389, Newcastle, U.K., June 1990.

[40] W. Xiaoqing and K. Kinoshita: "A Testable Design of Logic Circuits under Highly Observable Condition", *Proc. ITC 1990*, pp. 955-963, Washington, Sept. 1990.

[41] W. Xiaoqing and K. Kinoshita: "Test Pattern Generation for *k* - UCP Circuits", *Proc. CFTC-4*, pp. 119-125, Shanghai, Aug. 1991.

[42] W. Xiaoqing and K. Kinoshita: "Testable Designs of Sequential Circuits under Highly Observable Condition", *Proc. ITC 1992*, pp. 632-640, Baltimore, Sept. 1992.

[43] W. Xiaoqing and K. Kinoshita: "Testable Designs of Sequential Circuits under Highly Observ-

able Condition", *Trans. IEICE Trans. Inf. & Syst., Vol. E75-D, No. 3*, pp. 334-341, 1992.

[44] R. L. Wadsak: "Fault Modeling and Logic Simulation of CMOS and MOS Integrated Circuits", *Bell Syst. Tech. J., Vol. 57*, pp. 1449-1474, May-June, 1978.

[45] R. Chandramouli: "On Testing Stuck-Open Faults", *Proc. FTCS-13*, pp. 258-265, 1983.

[46] R. Davis: "Diagnostic Reasoning Based on Structure and Behavior", *Artificial Intelligence 24*, pp. 347-410, 1984.

[47] 日本電子顕微鏡学会関東支部編：*走査型電子顕微鏡－基礎と応用－*，共立出版，昭51．

[48] S. Murai, C. Tanaka, S. Nakamura, T. Ogihara, M. Terrai, and K. Kinoshita: "An Integated Computer Aided Design System for Gate Array Masterslice: Part 1 - Logic Reorganization System LORES-2", *Proc. DAC-18*, 1981.

[49] J. E. Hopcropft and J. D. Ullman: *Introduction to Automata Theory, Language, and Computation*, Addison-Wesley, 1979.

[50] J. Savage: *The Complexity of Computing*, Wiley, 1976.

[51] S. Muroga: *Logic Design and Switching Theory*, John Wiley & Sons, Inc, 1979.

[52] M. K. Reddy, S. M. Reddy, and P. Agrawal: "Transistor Level Test Generation for MOS Circuits", *Proc. DAC-22*, pp. 825-828, 1985.

[53] E. J. McClusky: "Verification Testing - A Pseudoexhastive Test Technique", *IEEE Trans. Comput., Vol. C-33, No. 6*, pp. 541-546, 1984.

[54] J. Savir: "Syndrome Testing of Syndrome-Untestable Combinational Circuits", *IEEE Trans. Comput., Vol. C-30, No. 8*, pp. 606-608, 1981.

[55] 岡本務，樹下行三："半加算器アレイの故障検査について"，*信学論 C, Vol. 54-C, No. 5*, pp. 362-369, 1971.

[56] 松田潤一，樹下行三："半加算器アレイの故障検査の一般化について"，*信学論 D, Vol. 56-D, No. 3*, pp. 140-145, 1973.

[57] H. Fujiwara, K. Kinoshita and H. Ozaki: "Universal Testsets for Programmable Logic Arrays", *Proc. FTCS-10*, pp. 137-142, 1980.

[58] K. K. Sakuja, K. Kinoshita, and H. Fujiwara: "An Easily Testable Design of Programmable Logic Arrays for Multiple Faults", *IEEE Trans. Comput., Vol. C-32, No. 11*, pp. 137-142, 1983.

[59] K. S. Ramanatha, N. N. Biswas: "A Design for Complete Testability of Undetectable Cross-point Faults in Programmable Logic Arrays", *IEEE Trans. Comput., Vol. C-32, No. 6*, pp. 551-557, 1983.

[60] R. C. Minnick: "Cutpoint Cellular Logic", *IEEE Trans. Comput., Vol. EC-13, No. 6*, 1964.

[61] W. Xiaoqing, N. Itazaki, and K. Kinoshita: "Efficient Methods for Guided-Probe Diagnosis", *Trans. IEICE Trans. Inf. & Syst.*,（投稿中）．

[62] S. Kochan, N. Landis, and D. Monson: "Computer-Guided Probing Techniques", *Proc. ITC 1981*, pp. 253-268, 1981.

[63] M. Melgara, M. Battu, P. Garino, J. Dowe, Y. J. Vernay, M. Marzouki, and F. Boland: "Automatic Location of IC Design Errors Using E-Beam System", *Proc. ITC 1989*, pp. 898-907, 1989.

[64] 白川千洋，久慈憲夫："信号注入シミュレーションによる EBT ガイディドプローブ"，*日本学術振興会第 132 委員会第 117 回研究会資料集*，pp. 59-64, 1991.

[65] H. Cox and J. Rajski: "A Method of Fault Analysis for Test Generation and Fault Daignosis", *IEEE Trans. Computer-Aided Design, Vo. 7, No. 7*, pp. 813-833, 1988.

[66] H. Nijima and A. Hu: "Aavanced E-Beam Testing", *日本学術振興会第 132 委員会第 116 回研究会資料集*，pp. 68-75, 1990.

[67] 山口昇，坂本隆，西岡寿，真島敏幸，佐藤司，品田博之，鈴木寛，戸所秀男，山田理："論理用電子ビームテスタによるガイディドプローブ法の検討"，*日本学術振興会第 132 委員会第 117 回研究会資料集*，pp. 53-58, 1991.

[68] 大窪和生，手操弘典，伊藤昭夫："EB テスタにおける LSI 配線への EB 自動位置決め方式の検討"，*日本学術振興会第 132 委員会第 117 回研究会資料集*，pp. 153-158, 1991.

[69] Y. Tokunaga, T. Nakamura, M. Seyama, P. Fazekas, H. Feuerbaum, and J. Frosien: "Aavanced E-Beam Testing Part I", *日本学術振興会第 132 委員会第 117 回研究会資料集*，pp. 80-85, 1991.

[70] 山田輝彦，中村芳行："組合せ回路における単一縮退故障の一診断法"，*電子情報通信学会論文誌 DI, Vol. J74, No. 11*, pp. 774-780, 1991.

[71] F. Brglez and H. Fujiwara: "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translaor in Fortran", *Proc. ISCAS 1985*, pp. 705-712, 1985.

[72] F. Brglez, D. Bryan and K. Kozminski: "Combinational Profiles of Sequential Benchmark Circuits", *Proc. ISCAS 1989*, pp. 1929-1934, 1989.

研究発表一覧表

学会誌発表論文

1. 樹下行三，温暁青，スダーカ．M．レディ
   "全可観測な環境での NAND 回路の故障診断"
   電子情報通信学会論文誌 DI, Vol. J73, No. 2, pp. 245-252, 1990.
2. Wen Xiaoqing and Kozo Kinoshita
   "Testable Designs of Sequential Circuits under Highly Observable Condition"
   IEICE Trans. on Inf. & Syst., Vol. E75-D, No. 3, pp. 334-341, 1992.
3. Wen Xiaoqing and Kozo Kinoshita
   "Testable Design of Logic Circuits under Highly Observable Condition"
   IEEE Trans. on Comput., Vol. 41, No. 5, pp. 654-659, May 1992.
4. Wen Xiaoqing, Noriyoshi Itazaki, and Kozo Kinoshita
   "Efficient Methods for Guided-Probe Diagnosis"
   IEICE Trans. on Inf. & Syst.,（投稿中）.

国際会議発表論文

5. Kozo Kinoshita, Wen Xiaoqing and S. M. Reddy
   "Diagnostic Testing of NAND Circuits under Totally Observable Condition"
   Proc. of JFTCS'89, pp. 69-74, Chongqing, China, June 1989.
6. Wen Xiaoqing and Kozo Kinoshita
   "Fault Detection and Diagnosis of k - UCP circuits under Totally Observable Condition"
   Proc. of FTCS-20, pp. 382-389, Newcastle, U.K., June 1990.
7. Wen Xiaoqing and Kozo Kinoshita
   "A Testable Design of Logic Circuits under Highly Observable Condition"
   Proc. of ITC'90, pp. 955-963, Washington D. C., U. S. A., Sept. 1990.
8. Wen Xiaoqing and Kozo Kinoshita
   "Testable Designs of Sequential Circuits under Highly Observable Condition"
   Proc. of ITC'92, pp. 632-641, Baltimore, U. S. A., Sept. 1992.

研究会発表論文

9. 梶原誠司，温暁青，板崎徳禎，樹下行三
   "可観測な環境でのテストパターン生成について"
   情報処理学会設計自動化研究会資料，88-DT-44, pp. 35-42, 1988.
10. 梶原誠司，温暁青，板崎徳禎，樹下行三
    "可観測な環境でのテストパターン生成について"
    第１９回ＦＴＣ研究会資料集, 1988.

11. 樹下行三，温暁青
"全可観測な環境での論理回路の故障診断について"
第２０回ＦＴＣ研究会資料集, 1989.
12. 樹下行三，温暁青
"全可観測な環境での論理回路の故障診断について"
信学技報, Vol. 89, No. 456, pp. 44-48, 1989.
13. 温暁青，樹下行三
"全可観測な環境での順序回路の故障検査について"
信学技報, Vol. 90, No. 238, pp. 9-16, 1990.
14. 温暁青，樹下行三
"全可観測な環境でのスキャンパスを有する順序回路の故障検査について"
信学技報, Vol. 90, No. 238, pp. 9-16, 1990.
15. Wen Xiaoqing and Kozo Kinoshita
"Test Pattern Generation for *k* - UCP Circuits"
Proc. of CFTC-4, pp. 119-125, Shanghai, China, Aug. 1991.