

卒業研究報告

題目

可制御性向上を考慮した故障検出強化法に関する研究

指導教官

高橋寛教授

王森レイ講師

報告者

青野智己

令和2年1月29日提出

愛媛大学工学部情報工学科計算機システム講座

目次

第 1 章 概要	1
第 2 章 序論	3
第 3 章 マルチサイクルテストとその課題	7
3.1 マルチサイクルテスト	7
3.2 故障信号消失問題	8
3.3 故障検出能力低下問題	10
第 4 章 制御テストポイント (CP) 挿入技術	11
4.1 故障検出能力低下問題への対策方針	11
4.2 CP の制御方法	12
第 5 章 CP 選定方法	13
5.1 FF に対する CP 挿入方法	13
5.2 組合せ回路内部に対する CP 挿入方法	14
第 6 章 評価実験	20
6.1 評価実験方法	20
6.2 評価実験結果	22
第 7 章 まとめ	23
謝辞	24

参考文献	25
------	----

発表論文リスト	26
---------	----

第 1 章

概要

近年、自動車の安心・安全を保障するための先進自動運転システムの開発が急速に進んでいる。先進自動運転システムの実現に向けて、複雑な処理をリアルタイムに制御するための車載システムオンチップ (SoC) の高度な安全性が求められている。そこで、自動車の安全性を評価するための機能安全規格 ISO26262 が制定された。車載 SoC を機能安全規格に対応させるためには、パワーオンセルフテスト (POST) が必要になる。POST は車載システムの起動時に組み込み自己テスト (LBIST) を用いて、被検査回路内の物理的な故障を検出するフィールドテスト技術である。POST で ISO26262 で定められている最も厳しい安全要求レベル (ASILD) を満たすためには、短いテスト実行時間 (およそ数十 ms) で高い故障検出率 (90%以上) を達成しなければならない。そのために LBIST を用いて少数パターンで 90%以上の故障検出率を得るためのテスト技術が必要である。

制約を満たすためのテスト技術としてマルチサイクルテストが提案されている [1]。マルチサイクルテストは LBIST のスキャンテスト方式において、キャプチャサイクルを複数回実行する方法である。通常のスキャンテストに比べて、一回のスキャンイン動作で故障を検出する機会が多く得られるため、テスト実行時間の短縮ができる。しかしながら、マルチサイクルテストでは故障検出率の向上を妨げる二つの問題が観測された。一つは、長い伝搬経路のために故障信号が最終観測点まで到達されずに消失してしまう故障信号消失問題である。この問題は、文献 [2] の中間観測技術により解決された。もう一つの問題は、サイクルを重ねるにつれて FF または組合せ回路部の論理値が固定

(0 または 1) になることにより, 新たな故障の検出が困難になる故障検出能力低下問題である.

本研究の目的は, 機能安全のための制約を満たすために, マルチサイクルテストの故障検出能力を向上させることである. そのために本研究の目標を故障検出能力低下問題に対する解決方法の提案および提案方法の実装, 評価と設定する.

故障検出能力低下問題への対策として, 被検査回路内部に直接論理値を割り当てる制御テストポイント (CP) を挿入することを考える. 制御テストポイントの挿入により, サイクルを得たキャプチャパターンのランダム性や故障信号の伝搬効果を高め, 故障検出能力の向上を期待する. しかしながら, マルチサイクルテストに適用可能な CP 技術が存在していない. そのため, 本論文では, マルチサイクルテスト向けの CP を実現するために, テストポイントの制御方法および CP 挿入箇所の選定アルゴリズムを提案する.

本稿は以下のように構成される. 1 章では, 研究の背景および目的について述べる. 2 章では, 用語の説明を行う. 3 章では, マルチサイクルテストとその課題について述べる. 4 章では, 問題への対策方針としての CP 技術について述べる. 5 章では, CP 挿入点の選定方法について述べる. 6 章では, 提案法による評価実験結果を示す. 7 章では, 本研究のまとめを述べる.

第 2 章

序論

本章では, 本論文で対象とする用語の定義について述べる [3][4].

縮退故障 (stuck-at fault)

半導体の物理的欠陥を表現するための故障モデルの一つである. 入力信号とは無関係に常に信号線の値が固定される故障を縮退故障という. 論理値が 0 で固定される故障を 0 縮退故障, 1 で固定される故障を 1 縮退故障という. 本稿では, 取り扱う故障モデルは縮退故障とする.

テスト

回路に入力パターンや入力系列を加え, それに対する回路の出力パターンや出力応答系列を観測し, 回路に故障が存在するか否かを調べることを故障検出という. さらに, 回路に故障が存在することが分かったとき, それがどのような故障であるかを調べることを故障診断という. 故障検出することおよび故障診断することをテストと呼ぶ. 本稿では, 故障診断については取り扱わない.

故障検出率

テストパターン (系列) 集合で, 最初に想定した故障のうちどれだけの故障が検出されるかを示す比率 (%) を故障検出率と呼ぶ.

$$\text{故障検出率} = \frac{\text{検出された故障数}}{\text{総故障数}} \times 100(\%)$$

故障シミュレーション

故障回路と正常回路に対して, テストを加えた場合の回路動作のシミュレーションを行い, 与えられたテストパターン (系列) がどの故障を検出するかを調べるものである.

LBIST(Logic Built In Self Test)

POST で用いられる LBIST はスキャンテストを基に設計されており, 図 2.1 のような構成を持つ. テストパターン発生回路としては, 線形フィードバックシフトレジスタ (LFSR:linear feed back shift register) が用いられる. LFSR では, すべて 0 のパターンを除くすべてのパターンを疑似ランダム的に発生することができる. テストパターンに対する出力応答パターンをすべて正しい期待値と比較する方法では, 期待値を蓄えるための膨大なメモリが必要となり, 多くの時間がかかるという欠点がある, このため, LBIST では出力応答系列を LFSR など圧縮し, 最後に残る値 (シグネチャ) と正しい期待値と比較してテストを行う.

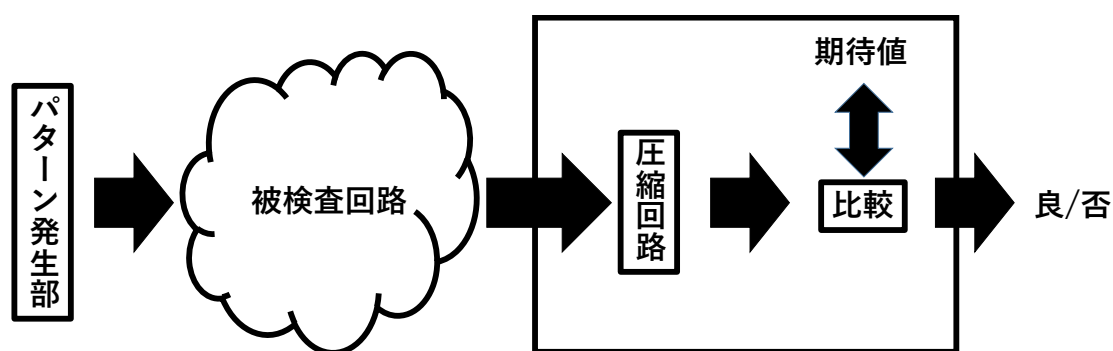


図 2.1. LBIST 構成

スキャンテスト

今日の大規模集積回路のテスト容易化設計にはスキャン設計が適用される。図2.2にスキャン設計の例を示す。スキャン設計では、すべてのFFが、スキャンモード時にシフトレジスタとして動作するように設計する。赤色で示されている部分がスキャンのために付加された回路である。mode=0とすると、各FFの入力は組合せ回路部の出力と接続され、通常動作（本来の動作）をする。mode=1とすると、SI(Scan-In)を入力、SO(Scan-Out)を出力とするシフトレジスタとして動作する。この回路により、すべてのFFの値は組合せ回路部とは独立して自由に制御・観測できるようになる。

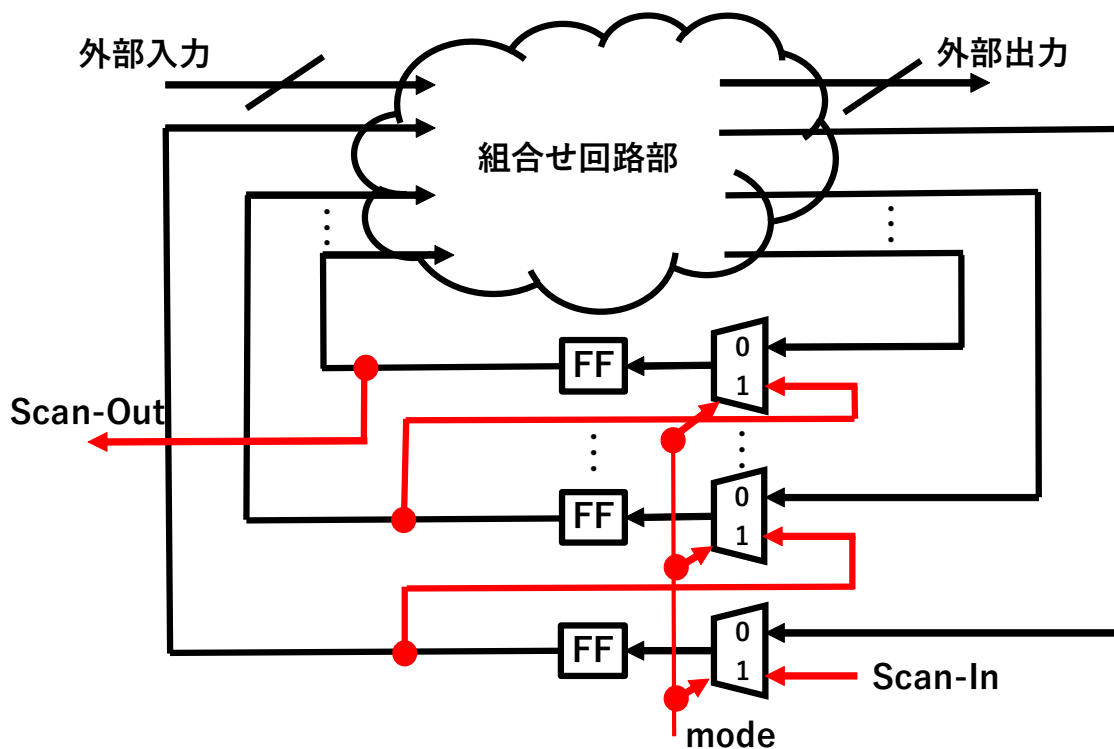


図 2.2. スキャンテスト設計

このスキャン設計された回路に対するテスト実行は次のようになる。ここでFF数をNとする。mode=1としてSIからNビットの系列を入力し、FFの値を設定する（スキャンイン）。mode=0として、外部入力からテストパターンを入力し、1クロックを進める

(キャプチャ). 組合せ回路部の出力の一部は外部出力で観測され, 残りは FF に記憶される. mode=1 として, N クロックを進める (スキャンアウト). FF に記憶された組合せ回路部の出力を SO から N ビットの系列として観測する.

第3章

マルチサイクルテストとその課題

本章では, スキャンテストの応用であるマルチサイクルテストとその課題について説明を行う.

3.1 マルチサイクルテスト

スキャンテストでは, FF 数 N に比例したテスト実行時間を要することが課題である. そこで, キャプチャクロックを複数回実行するマルチサイクルテストが提案された. 図 3.1 にマルチサイクルテストを適用した組合せ回路の時間展開図を示す. マルチサイクルテストとは, キャプチャされたパターン系列を次のテストの入力として再利用する方法である. サイクル数 $M=3$ とした場合のテスト実行例は次のようになる. スキャンイン動作で (110) が入力される. 1 サイクル目でキャプチャされたパターン系列 (001) を入力として 2 サイクル目が実行される. 2 サイクル目でキャプチャされたパターン系列 (101) を入力として 3 サイクル目が実行される. 3 サイクル目でキャプチャされたパターン系列 (101) をスキャンアウトすることにより観測する.

マルチサイクルテストでは, 1 回のスキャンイン動作で複数個のテストパターンでのテストを実行することができる. そうすることで, 多くの故障を励起する機会が与えられ, 1 回のスキャンイン動作に対する故障検出能力の向上ができるため, テスト実行時間の短縮が期待される. しかしながら, マルチサイクルテストでは故障検出能力が低下してしまう二つの問題が観測された.

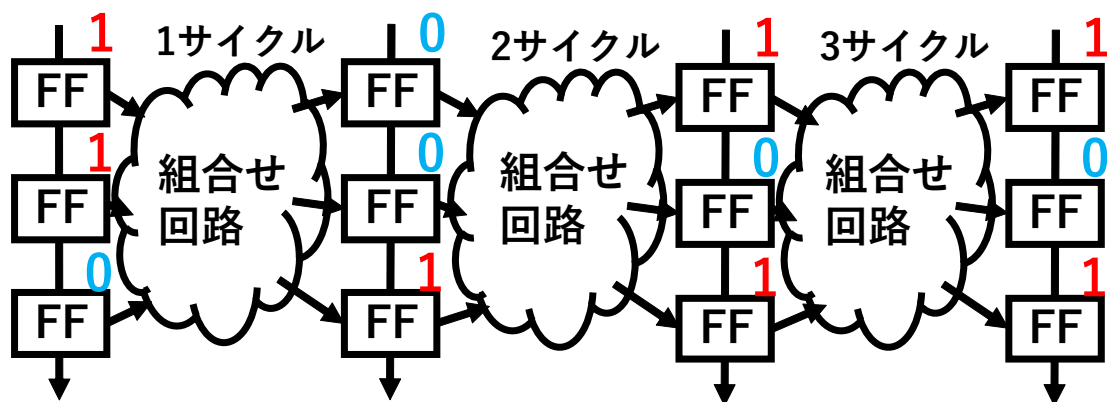


図 3.1. マルチサイクルテスト

3.2 故障信号消失問題

問題の一つは、図 3.2 に示される故障信号消失問題である。図は 1 サイクル目で励起された故障信号は、2 サイクル目を経由し 3 サイクル目に突入したところで消失してしまうことを表している。この場合、観測点である 3 サイクル目の出力部まで到達していないため、故障検出をすることができない。このように、励起された故障が長い伝搬経路のために、観測点まで到達できずに消失してしまうことで故障検出率が向上しない問題がある。

この問題は文献 [2] の中間観測技術により解決された。中間観測技術とは図 3.3 のように一部の FF を毎サイクルで観測することにより、故障信号が消失するまでに故障検出を可能にしたものである。

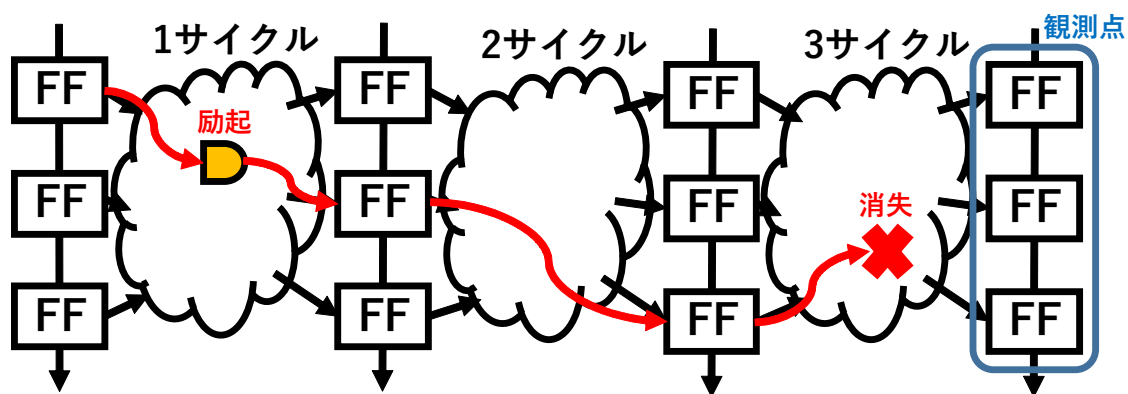


図 3.2. 故障信号消失問題

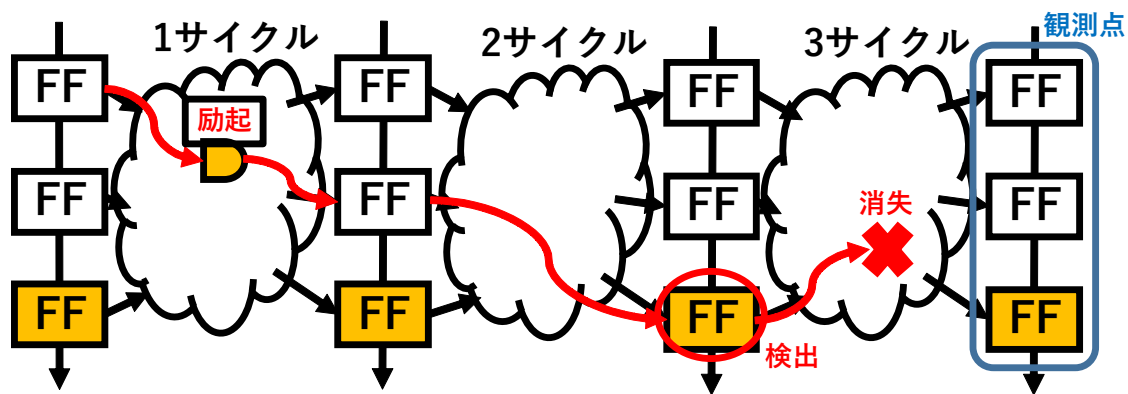


図 3.3. 中間観測技術

3.3 故障検出能力低下問題

文献 [5] ではマルチサイクルテストにおいて、キャプチャサイクルを増やすことで被検査回路の内部状態が機能動作に近づくことが報告されている。それにより、生じる故障検出能力低下問題を次のような問題として考える。図 3.4 は被検査回路に対してマルチサイクルテスト (3 サイクル) を適用した例を示す。この例では、スキャンインにて初期値ベクトル (110) が与えられる。次に、1 回目のキャプチャで得られたキャプチャベクトル (011) を次のサイクルの入力として再利用する。以下、同様にしてキャプチャ動作が 2 回目と 3 回目とそれぞれ実行される。ここで、2 回目以降のキャプチャ動作時に回路内部の AND ゲートの入力値が 0 に固定する場合を考える。それによって、AND ゲートの出力値は 0 となり、その影響は FF や次のサイクルの論理ゲートまでにおよぶことになる。この影響を受けた論理ゲートの入力コーン内にある故障信号は伝搬されることはないため、2 サイクル目以降の故障検出能力は低下してしまう。

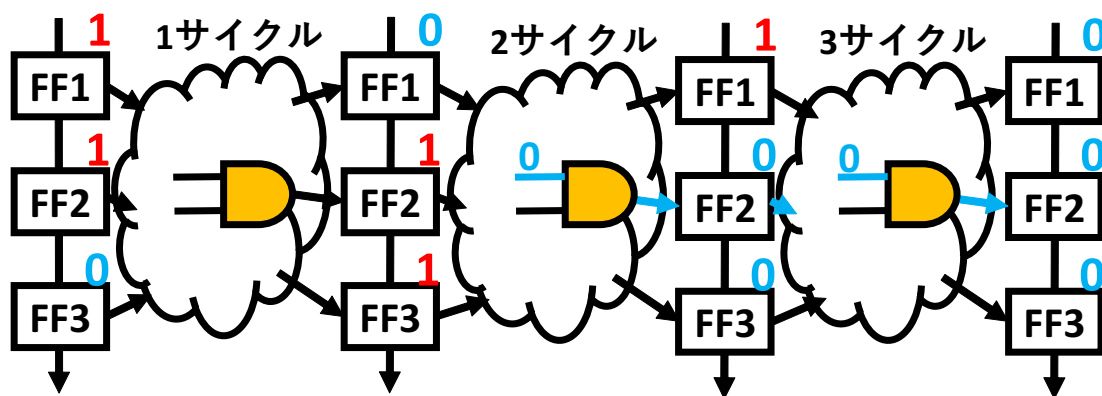


図 3.4. 故障検出能力低下問題

第 4 章

制御テストポイント (CP) 挿入技術

本章では, マルチサイクルテストにおける故障検出能力低下問題への対策方針として CP 挿入技術を提案する.

4.1 故障検出能力低下問題への対策方針

3 章で説明したように, マルチサイクルテストは, キャプチャサイクルを経ることで被検査回路の論理ゲートの入出力部が固定値になり, 新たな故障の検出が困難になる問題がある. この問題の対策案は, 固定値となることで故障信号の伝搬や励起を阻害している箇所に直接論理値を割り当てる制御テストポイント (CP) を挿入することである. CP の役割は故障信号の励起と故障信号の伝搬の二つであると考え. 二つの効果をそれぞれ確認するために, CP の挿入点を FF と組合せ回路内部に分けて考える.

FF に CP を挿入する場合は, 制御後のサイクルにおけるキャプチャパターンのランダム性を向上させることができる. 0/1 縮退故障に対して固定値 0/1 が印加され続ける箇所に対して, 制御した論理値 1/0 を割り当てることにより, 故障の励起を促し, 故障検出能力の向上を期待する. そのために, 出力に大きな論理影響領域を持つ FF を CP 挿入点に選択する必要がある.

組合せ回路内部に CP を挿入することにより, 固定値となることで回路内の故障信号の伝搬を阻害する箇所に直接論理値を割り当てることことができる. 論理値が固定化となる箇所に, 非制御値 (AND ゲートの入力に論理値 1, OR ゲートの入力に論理値 0) を設

定することで, 故障伝搬を促し, 故障検出能力の向上を期待する.

これらの方法を実現するために, CP の制御方法および CP 挿入箇所 (FF と組合せ回路内部) の選定アルゴリズムを提案する.

4.2 CP の制御方法

各キャプチャサイクルで FF を制御するために 2 種類の制御構造を提案する. 図 4.2 に示す 2 種類の制御構造を選定点に適用する.

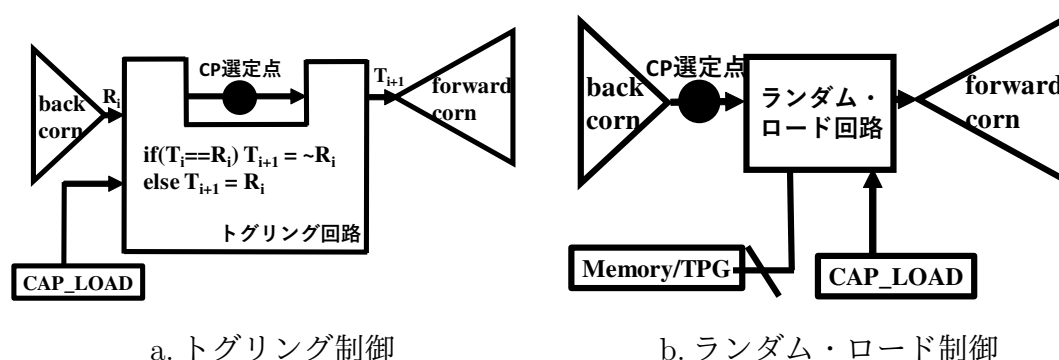


図 4.1. 制御構造

図 4.1.a の制御方法は, 選定した信号線にトグル回路を追加することで, サイクル毎に選定信号線の値をトグルさせるトグリング方法である. キャプチャモードでは, 現在の状態 (T_i :現在のキャプチャサイクルで適用されたテストパターン) と次の状態 (R_i :現在のキャプチャサイクルで適用されたテストパターンのレスポンス) を比較して, トグルが発生しているかを確認する. トグルがない場合, トグル制御回路は T_{i+1} にトグルを生成し R_i の反転値を次のサイクルに伝搬させ, そうでなければ次のサイクルに現在の R_i を伝搬させる. 外部制御信号 (CAPLOAD) が 1 のときにトグリングを有効にし, 0 のときに通常のキャプチャ動作を実行する.

図 4.2.b の制御方法は, 擬似ランダムベクトルを選定信号線に直接印加するランダム・ロード方法である. 選定信号線にランダム・ロード回路が挿入され, 外部制御信号が回路を制御し, 選定点の値または擬似ランダムベクトルのいずれかを選択して次のサイクルに伝搬させる. 擬似ランダムベクトルはメモリから供給するかあるいはオンチップのテストパターンジェネレータ (TPG) によって生成する.

第 5 章

CP 選定方法

本章では,CP 挿入箇所の選定方法について述べる. CP の選定方法は,FF と組合せ回路内部それぞれに限定した方法である.

5.1 FF に対する CP 挿入方法

より大きな論理出力コーン (FF に出力に接続された論理影響領域) を持つ FF を制御することにより,多くの故障を励起する可能性が高くなる. 従って,FF の論理出力コーン内の構造解析を行うことにより,FF の選定を行う. FF の構造解析評価値は以下の 5 つを定義する.

P1. FF が持つ論理影響コーンの最大の深さ

P2. FF が持つ論理影響コーンの最大幅

P3. FF が持つ論理影響コーンに存在する分岐の総数

P4. FF と PO の最長経路

P5. FF が持つ論理影響コーンに存在するゲート数

図 5.1 に P1~P5 の計算例を示す. 各 FF に対して順方向トレース処理によって P1~P5 の値を導出する. 得られた評価値を入力として多岐基準決定法 [6] のアルゴリズムを実行することで,より大きな論理影響領域を有する FF が決定される.

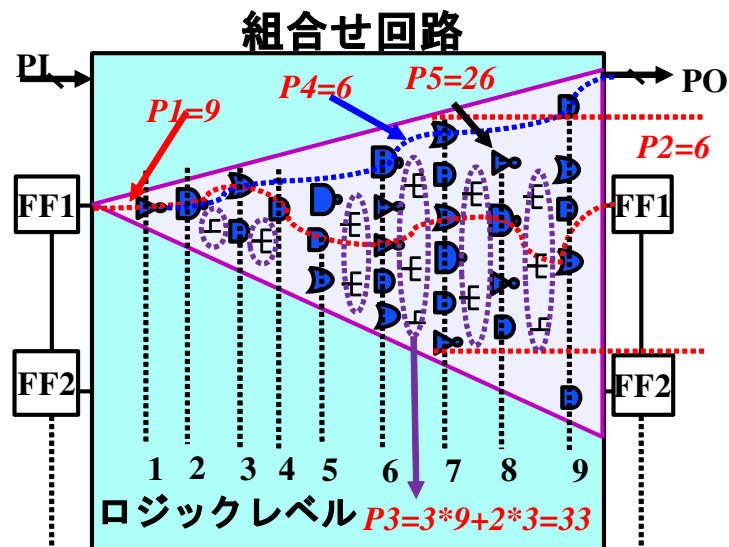


図 5.1. 構造解析評価値

5.2 組合せ回路内部に対する CP 挿入方法

論理ゲートの入力が制御値（例:AND ゲートに対して 0,OR ゲートに対して 1）で固定になっている箇所に対して CP により非制御値を与えることは、故障信号を伝播させることに役立つ。このような箇所を組み合わせ回路内部から特定するために以下の評価値を定義する。なお、 l は論理ゲートの入出力信号線である。

$p0(l)/p1(l)$: l における確立ベースのテストビリティ評価尺度 (COP) の 0/1 可制御性

$f0(l)/f1(l)$: l を 0/1 に設定することにより、 l から出力側（外部出力または FF）までの到達可能論理回路部に論理値が固定値になるゲートの数

$c(l)$: l を制御することによる論理値が固定化されているゲートへの寄与度

以下、各評価値の計算方法を説明する。

A. 確率ベーススタビリティ評価尺度 (p_0/p_1)

信号線 l が 0 または 1 にどの程度なりやすいかを調べるために, $p_0(l)/p_1(l)$ を計算する. $p_0(l)/p_1(l)$ は COP ベースの可制御性の計算方法 [7] に従う. COP の可制御性計算は被検査回路の FF の初期値を 0.5 に設定し, 後続する l に対して $p_0(l)/p_1(l)$ を計算する. 本稿では, マルチサイクルテスト向けの CP 挿入点の選定を行うため, FF の初期値を変更する. FF の初期値は, キャプチャサイクル数回展開した被検査回路における COP の平均値とする. マルチサイクルテストにおける COP の計算例を図 5.2 に示す.

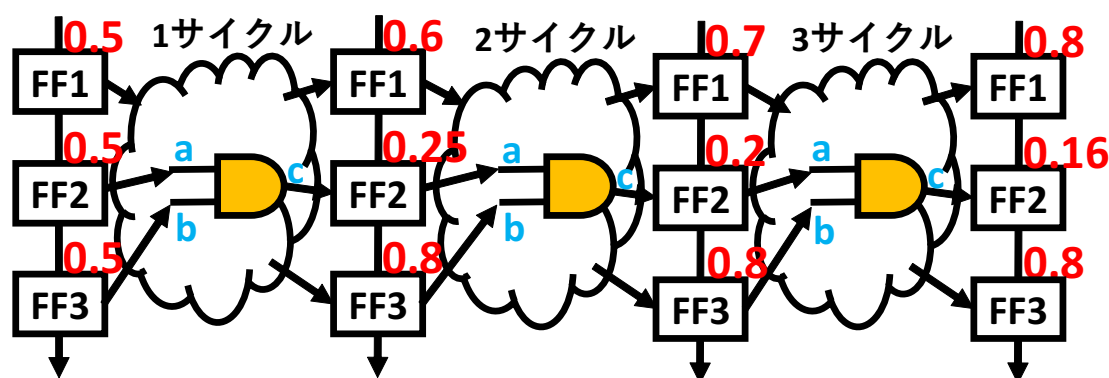


図 5.2. マルチサイクルテストにおける cop の計算例

図 5.2 は被検査回路を時間展開した図であり, 各 FF の 1 可制御性を示している. FF2 および FF3 はそれぞれ AND ゲートの入力 a と b に接続している. また, AND ゲートの出力 c は FF2 に接続する. 1 サイクル目には, 可制御性の初期値として 0.5 が与えられる. 従って, 1 サイクル目における c の 1 可制御性は $0.25 (= 0.5 * 0.5)$ となり, 2 サイクル目の初期値として用いられる. 同様にして 2 サイクル目, 3 サイクル目が計算される. 結果として, FF2 の初期値は $0.2 (= (0.25 + 0.2 + 0.16) / 3)$ となる.

図 5.3 は, 図 5.2 の結果をもとに FF の初期値を設定した時の被検査回路である. 図 5.3 より $p_1(a)=0.2, p_1(b)=0.8, p_1(c)=0.16$ が求められる.

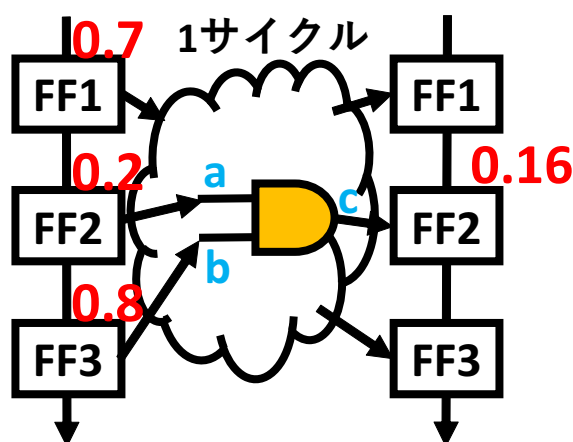


図 5.3. cop 計算例

B. 構造解析による固定化ゲートの評価値 (f_0/f_1)

信号線 l を 0(1) に設定することによって, その信号線 l が出力側 (外部出力または FF) までの到達可能論理回路部にいくつかのゲートの出力線の論理値が固定値になるかを調べるために, $f_0(l)/f_1(l)$ を計算する. 図 5.4 に示すゲート構造例で計算例を説明する.

G1 の入力線 a の論理値を 0 に設定する場合を考える. 0 は AND ゲートの制御値であるため b の論理値に影響されることがなく, 信号線 c, d, e は 0 となる. OR ゲートの制御値は 1 であるため, G2 はもう一つの入力値により論理値が決定される. G3 は e の論理値が 0 のため, もう一つの入力値に関係なく値が 0 となる. 従って, a を 0 とした場合, G1 と G3 の論理値が他の入力に関係なく, 決定されるため, $f_0(a)=2$ となる. 一方で, a を 1 に設定した場合, b の入力値に依存するため, 信号線 c, d, e はドントケアとなる. 従って, $f_1(a)=0$ である.

同様にして, 残りの入出力線 b, c, d, e に対して f_0/f_1 を計算した結果を 5.1 に示す.

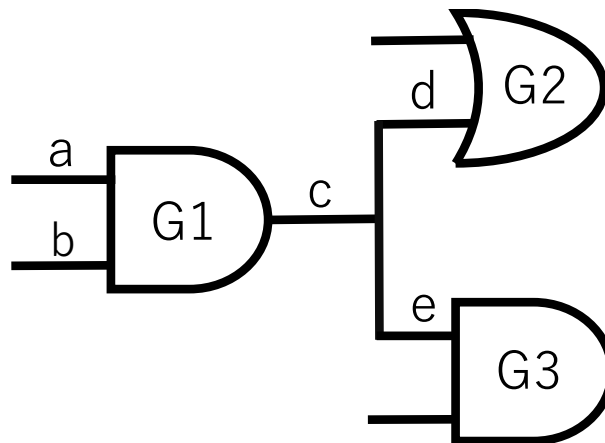


図 5.4. ゲート構造例

表 5.1. 構造例の $f_0(l)/f_1(l)$

l	f_0	f_1
a	2	0
b	2	0
c	1	1
d	0	1
e	1	1

C.CP 制御による固定化ゲート削減への寄与度 (c)

信号線 l の $p0/p1$ および $f0/f1$ を用いて, CP 制御することにより固定化するゲートをどの程度減らすことができるかを評価値とするために c_l を以下のように定義する.

$$c(l) = -((0.5 - p0(l)) * f0(l) + (0.5 - p1(l)) * f1(l))$$

図 5.4 において, $p0(a)=0.9, p1(b)=0.5$ であるケースを考える. この時に, 信号線 a, b, c, d, e の $p0(l)/p1(l), f0(l)/f1(l), c(l)$ は表 5.2 のように計算される.

表 5.2. 構造例の $c(l)$

l	$f0$	$f1$	$p0$	$p1$	c
a	2	0	0.9	0.1	0.8
b	2	0	0.5	0.5	0
c	1	1	0.95	0.05	0
d	0	1	0.95	0.05	0.05
e	1	0	0.95	0.05	0.45

表 5.2 の例によると, 信号線 a の論理値を 0 にすると, a から到達可能な論理回路部には 2 個のゲートが固定化される. 一方, 信号線 a の論理値が 0 になる確率は $p0(a)=0.9$ であるため, 信号線 a から出力までの論理回路部には多くのテストで 2 個のゲートが固定化される可能性が高い. このような特徴を持つ信号線の論理出力コーンは故障信号の伝搬を妨げるゲートを多く持つため, 制御した方がよい ($c(l)$ が大きい) という結果になる. 信号線の c 値が高いほど, その信号線に CP を挿入した方がよいと考える.

これらの評価値を用いて CP を選定する手順 5.1 を提案する. 3~8 の処理で候補 CP リストを作成するのには計算量削減の目的がある. 12 と 14 の処理で影響領域の重なり判定をすべての信号線に対して実行するためには, 膨大な計算量が必要になる. そのため, 論理値が固定になりやすい信号線に限定して選定を行う.

アルゴリズム中で用いられる α ($0 \leq \alpha \leq 1$) および β ($0 \leq \beta \leq 1$) はユーザ指定可能な閾値である. 例えば, α に対しては, $p0(l)/p1(l)$ の最大値から 0.1 以内の範囲にある l を候補とするために $\max(p0(l), p1(l)) - 0.1$ を設定する. β に対しては, すでに選定し

た CP リストの出力コーンと 50%以上の重なりがある点を選定しないようにするために 0.5 を設定する.

疑似コード 5.1. 選定アルゴリズム

INPUT

論理回路: ゲート数=N, M: 選定する CP 数
 α : p0 と p1 の偏り度の閾値, β : 重なり of the rate の閾値

DATA

A = $\{\phi\}$: 選択した CP リスト
 B = $\{\phi\}$: 一時的に保存する候補 CP リスト
 rate_overlap(A, B[i]): 選定された箇所全ての影響領域と B[i] の影響領域との重なり of the rate を出力する

OUTPUT

優先度が高い順にソートされた CP リスト集合 A

Preprocessing:

入出力線 l に対して p0(l), p1(l), f0(l), f1(l), c(l) を計算

Selection Processing:

```

1  i = 0, A =  $\{\phi\}$ 
2  for i = 1 to M do
3    B =  $\{\phi\}$ , B.length = 0, selection_flag = 0
4    for j = 1 to N do /*j=信号線番号*/
5      if (p0(j) >  $\alpha$ ) || (p1(j) >  $\alpha$ ) then
6        B  $\leftarrow$  j, B.length++
7      end if
8    end for
9    Sorting B in descending order by c
10   for k = 1 to B.length do
11     if rate_overlap(A, B[k]) <  $\beta$  then
12       A[i]  $\leftarrow$  B[k]
13       p0(A[i]) = 0.5, p1(A[i]) = 0.5
14       A[i] の前方領域の p0( $l$ ), p1( $l$ ), c( $l$ ) を更新
15       selection_flag = 1
16       break
17     end if
18   end for
19   if selection_flag == 0
20     return A
21   end if
22 end for
23 return A

```

第 6 章

評価実験

本章では, 提案手法と従来手法の効果を故障シミュレーションにより評価する.

6.1 評価実験方法

評価実験では, 故障検出率が 90%以上を達成した時点でのスキャンインパターン数を評価する. キャプチャサイクルを 10 サイクルとする縮退故障テストを実装し, ISCAS89 および ITC99 のベンチマーク回路を用いて内製の故障シミュレータで提案手法を評価した. 16 ビット LFSR($X_{16}+X_{15}+X_{13}+X_4+1$) を用いて最大 100K パターンを生成する. なお, 外部出力は観測しない. FF に対して選定した CP 数と組合せ回路内部に対して選定した CP 数はそれぞれ, FF 総数の 10%, ゲート総数の 1% である. 本実験で用いたベンチマーク回路で選択される CP 数を表 6.1 に示す. 制御方法は 4.2 節で説明したトグリング制御とランダム・ロード制御である. さらに文献 [2] で提案した中間観測テスト機構を FF の 20% 個に対して実装した.

表 6.1. 選定した CP 数

回路	CP 数	
	FF に挿入	ゲートに挿入
s9234	22	58
s13207	66	87
s15850	59	104
s38417	163	239
s38584	145	209
b14s	24	47
b15s	44	88
b20s	49	94

表 6.2. 90%を達成したテストパターン数

回路	制御なし	トグリリング		ランダム・ロード	
		FF_CTL	GT_CTL	FF_CTL	GT_CTL
s9234	(89.95)	68000	8050	51050	10400
s13207	11600	6350	5750	2650	3150
s15850	(89.01)	(88.93)	11300	(89.1)	63350
s38417	700	400	100	550	100
s38584	4000	4050	1650	1900	650
b14s	56250	(87.5)	1850	8400	500
b15s	4200	5850	750	4700	350
b20s	(89.68)	(89.41)	1500	12100	1900

6.2 評価実験結果

表 6.2 に目標検出率 (90%) を達成するために必要なテストパターン数を示す. 90% を達成できなかった手法には, 100k パターンで得られた故障検出率を記している. s9234 回路では, 制御なしの場合, 100k パターンで 89.95% の故障検出率を得ることができた. この回路に対して FF に CP を挿入した場合では, 68k パターンで 90% を達成 (32k パターンの削減) することができた. さらに, 組合せ回路内部に CP を挿入した場合では, 8k パターンで 90% を達成 (およそ 92k パターンの削減) をすることができた. 従って, s9234 回路では, 目標検出率を達成するためのテストパターンの削減が見られ, 故障検出能力の向上が確認できた. その他の回路では, FF を制御した場合に s13207 (トグリングとランダムロード), s38417 (トグリングとランダム・ロード), s38584 (ランダム・ロード), b14s (ランダム・ロード), b20s (ランダム・ロード) において故障検出能力の向上が見られた. 組合せ回路内部を制御した場合では, すべてのケースにおいて故障検出能力の向上が見られた.

FF を制御することは, 入力パターンのランダム性を向上させることになるため, 故障信号の励起効果が向上する. 組合せ回路内部を制御することは, 故障信号の伝搬効果が向上する. 回路内部を制御することにより, 多くのケースで故障検出能力の向上が見られたことから, 故障検出能力の向上には CP により故障信号を伝播させることが役に立つと考えられる. s13207 回路だけは, FF を制御したほうが良いという結果になったが, これは, 励起された故障信号が消失することなく観測点に到達できる構造であるために, 励起効果が役にたったと考えられる.

第 7 章

まとめ

マルチサイクルテストでは, キャプチャサイクルが増加するにつれて被検査回路の機能動作に近づき, いくつかの論理ゲートは固定化される. 固定化された論理ゲートは故障信号の伝搬を阻害するため, 故障検出能力を低下させる要因となる. このような問題を解決するために, 外部から直接論理値を割り当てる CP を被検査回路内部に挿入することを考える. 本論文では, マルチサイクルテストにおける CP の制御方法および挿入点の選定方法をそれぞれ提案した. さらに, 提案手法で選定した箇所に CP を挿入した回路に対してマルチサイクル故障シミュレーションを実施した. 結果からは, 提案手法により, 故障検出能力の向上が確認できた.

今後の課題として, 選定した CP の評価方法を考える. 現在は, 故障シミュレーションによる故障検出率で CP の評価を行っているが, 大規模回路では膨大な計算量が必要になるため, 短時間で故障検出率を推測する手法が必要である.

謝辞

本研究を進めるにあたり、懇篤な御指導、御鞭撻を賜りました本学高橋寛教授および王森レイ講師に深く御礼申し上げます。

本論文の作成に関し、詳細なるご検討、貴重な御教示を頂きました本学樋上喜信教授に深く御礼申し上げます。

また、審査頂いた本学阿萬裕久准教授に深く御礼申し上げます。

最後に、多大な御協力と貴重な御助言を頂いた本学工学部情報工学科計算機システム工学講座高橋研究室の諸氏に厚く御礼申し上げます。

参考文献

- [1] 山口 久登, 松藺 誠, 佐藤 康夫, 梶原 誠司, ”スキャンBIST におけるマルチサイクルテストと部分観測方式の提案と評価”, 電子情報通信学会技術研究報告, DC2010-28, pp.31-36, 2010-11
- [2] S. Wang, H. T. Al-Awadhi, S. Hamada, Y. Higami, H. Takahashi, H. Iwata, and J. Matsushim, ”Structure-Based Methods for Selecting Fault-Detection-Strengthened FF under Multi-cycle Test with Sequential Observation,” Proc. IEEE Asian Test Symposium, Hiroshima, pp. 209-214, Nov. 2016. DOI: 10.1109/ATS.2016.40
- [3] 藤原秀雄, ”コンピューターの設計とテスト”, 工学図書, 1990
- [4] 井上智生, ”半導体集積回路の欠陥とテスト技術”, 電子情報情報通信学会誌 Vol.103, No1, 2020
- [5] J. Rearick, ”Too Much Delay Fault Coverage is a Bad Thing,” Proc. Int ’ l Test Conf., Baltimore, MD, pp. 624-633, 2001. DOI: 10.1109/TEST.2001.966682
- [6] K. Yoon, ”A Reconciliation among Discrete Compromise Situation,” Journal of Operational Research Society, vol.38, no. 2, pp.277-286, 1987. DOI: 10.2307/2581948
- [7] F. Brglez, P. Pownall, and R. Hum, ”Applications of testability analysis: From ATPG to critical delay path tracing,” in Proc. ITC, Philadelphia, PA, USA, 1984, pp. 705-712.

発表論文リスト

- (1) 青野 智己, 矢野 良典, 王 森レイ, 樋上 喜信, 高橋 寛, ”マルチサイクルテストの故障検出率の低下を改善するためのキャプチャパターン制御法”, 平成 30 年度電気関係学会四国支部連合大会, 2018-9
- (2) 青野 智己, Hanan T.AI-Awadhi, 王 森レイ, 樋上 喜信, 高橋 寛, 岩田 浩幸, 前田 洋一, 松嶋 潤, ”マルチサイクルテストにおける故障検出強化のための FF トグル制御ポイントの選択法”, 電子情報通信学会技術報告, 2019-2
- (3) 青野 智己, 王 森レイ, 樋上 喜信, 高橋 寛, 岩田 浩幸, 前田 洋一, 松嶋 潤, ”車載システムの機能安全規格に準拠した POST の故障検出強化技術”, LSI とシステムのワークショップ 2019, 2019-5
- (4) 青野 智己, 王 森レイ, 樋上 喜信, 高橋 寛, ”マルチサイクルテストにおける故障検出低下問題の解析とその対策”, 令和元年度電気関係学会四国支部連合大会, 2019-9
- (5) 青野 智己, 中岡 典弘, 周 細紅, 王 森レイ, 樋上 喜信, 高橋 寛, 岩田 浩幸, 前田 洋一, 松嶋 潤, ”マルチサイクルテストにおける故障検出強化のためのテストポイント挿入法”, 電子情報通信学会技術報告, 2020-2