

卒業研究報告

題目

遅延故障に対するマルチサイクルテストの効果

指導教官

高橋寛教授

王森レイ助教

報告者

柳田 智也

平成 29 年 2 月 9 日提出

愛媛大学工学部情報工学科情報システム工学講座

目次

第 1 章 まえがき	1
第 2 章 準備	3
2.1 諸定義	3
2.1.1. 論理シミュレーション	3
2.1.2. 故障シミュレーション	3
2.1.3. テスト集合	4
2.1.4. 縮退故障	4
2.1.5. 遅延故障	4
2.1.6. フィールドテスト	5
2.2 スキャンベース BIST [3]	6
2.3 マルチサイクルテスト	7
第 3 章 遅延故障の中間観測マルチサイクルテスト	9
3.1 遅延故障のテスト方式	9
3.2 LoC 方式	10
3.3 回路構造の解析による中間観測 FF の選択手法	12
3.3.1. ゲート FF 接続複雑度による選択手法 (COMPLEX)	12
3.3.2. 時間展開を考慮した構造的な故障伝搬経路評価による選択手法	13
3.3.3. 分岐到達率評価による選択手法 (BRANCH)	15
第 4 章 実験結果・考察	17
4.1 実験環境	17

4.2 結果と考察	18
第 5 章 あとがき	25
謝辞	26
参考文献	27

第 1 章

まえがき

近年の微細化加工技術の発展に伴い,LSI の高集積化, 高速化が進んでいる. 高度自動化システムの発展は機能安全の重要性を高めたが,LSI の経年劣化による遅延故障が機能安全に影響を与え, 問題となっている.

経年劣化によって生じる遅延故障の早期検知の方法として,LSI の出荷後に行うフィールドテストがあげられる. フィールドテストとはシステムを実際に運用する中で, システムの起動時, あるいは待機時などに行うテストで, 厳しい時間制約がある. フィールドテストを実現する手法としてスキャンベース組み込み自己テスト (BIST : Built In Self Test) が存在する. スキャンベース BIST は設計手法の一種で, テストを容易にする手法である. 被検査回路内のフリップフロップ (FF : Flip-Flop) を数珠つなぎにして, スキャンチェーンと呼ばれるシフトレジスタを構成し,LSI チップ上に簡易的な LSI テスタ機能を有する回路を組込む手法である. 上記の設計を加えた回路は, スキャンチェーンのシフトによるスキャン入力と呼ばれるテストパターンの印加, 及びスキャン出力と呼ばれる結果の出力を行うシフトモードと, 印加したテストパターンを用いて回路の応答をスキャンチェーンにキャプチャするキャプチャモードという二つのモードを用いてテストを行う.

また, スキャンベース BIST では, 疑似乱数発生器が生成するランダム性のあるテストパターンがよく用いられる. しかしながら, 疑似乱数発生器が生成するテストパターンを用いた場合, 検出される故障に多数の重複が生じ, 高い故障検出率を得るまで膨大な時間がかかってしまう. フィールドテストは, 短い時間でテストを行わなければなら

ないため、膨大なテスト時間がかかってしまうという点は問題である。

この問題を解決するための手法として、マルチサイクルテスト [1] が提案されている。マルチサイクルテストとは、キャプチャ動作の際に、キャプチャサイクルを複数回連続で行い、キャプチャ毎に FF の値を観測するという手法である。キャプチャ動作を複数回行うことで、一度のテストで複数の内部状態を検査することができ、短時間で多くのテストを行うことが可能となる。しかしながら、キャプチャ動作を複数回行う中で、生じた故障の影響が途中から伝搬されなくなる問題がある。そのため、キャプチャサイクル毎に全ての FF を観測する、中間観測マルチサイクルテストが提案された。キャプチャサイクル毎に全ての FF を観測することで、キャプチャサイクルを複数回行う中で消失する故障を救済することが可能となる。しかしながら、全ての FF を観測することは、新たなゲートや配線を施すことを意味し、信号の遅延や面積オーバーヘッドの増大といった新たな問題を生むことになる。

中間観測の問題を解決する方法として、中間観測する FF を選択して、一部に限定するという部分観測手法が提案された。先行研究 [2] では、縮退故障を対象とした三つの中間観測 FF の選択法が提案された。しかしながら、経年劣化による遅延故障が多発し、縮退故障のみの故障検出率向上は機能安全の保障には不十分であると考えられる。

そこで、本研究ではスキャンベース BIST を用いた遅延故障検出を行う中間観測マルチサイクルテストにおいて、遅延故障検出率の向上が望める中間観測 FF の選択手法を研究する。具体的には、先行研究 [2] で提案した縮退故障向けの中間観測 FF の選択手法を遅延故障に用いて実験を行い、既存手法の有効性を評価、考察する。

本論文では、以下の構成をとる。二章では、本論文における用語の定義及び本研究の背景について述べる。三章では、遅延故障のテスト方式及びマルチサイクルテストにおける遅延故障のテスト法と先行研究 [2] で提案した中間観測 FF の選択手法を説明する。四章では、実験結果と考察を述べる。五章では、本研究全体のまとめと今後の課題について述べる。

第 2 章

準備

本章では, 本論文における用語と本研究背景について説明する. 本章の構成は以下の通りである. 2.1 節では, 故障検査に関する用語の定義を行う. 2.2 節では, スキャンベース BIST について述べる. 2.3 節では, スキャンベース BIST を用いたマルチサイクルテストについて述べる.

2.1 諸定義

2.1.1. 論理シミュレーション

論理回路の構造データを基に計算機上で仮想的な論理回路を再現し, 外部入力線にテストパターンを入力することで回路を動作させ, 論理回路の応答を出力させるプログラムを論理シミュレータと呼ぶ. 論理シミュレータを用いて入力パターンに対する演算結果を得る操作を論理シミュレートと呼ぶ.

2.1.2. 故障シミュレーション

論理シミュレータに任意の論理故障をつけ加え, 論理回路を構成させると故障した回路の挙動を調べることが可能となる. そして, 正常な回路と故障をつけ加えた回路に同一のテストパターンを入力し, 結果を比較することで回路の故障を検出するプログラムを故障シミュレータと呼ぶ. 故障シミュレータを用いて故障が検出するかどうかを調

べる操作を故障シミュレーションと呼ぶ。故障シミュレーションによって、検出された故障の数とつけ加えた故障の総数の割合である故障検出率を得る。

2.1.3. テスト集合

論理シミュレータ, 故障シミュレータを動作させる際に外部入力線に入力する値をテストパターンと呼び, 複数のテストパターンの集合をテスト集合と呼ぶ。

2.1.4. 縮退故障

論理回路内の信号線が特定の値に固定されてしまう故障を縮退故障と呼ぶ。特に, 値が 1 に固定される故障を 1 縮退故障, 0 に固定される故障を, 0 縮退故障と呼ぶ。

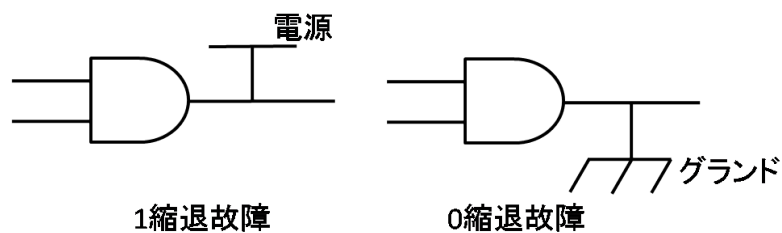


図 2.1. 縮退故障モデル

2.1.5. 遅延故障

論理回路内の素子や信号線における信号変化の伝搬遅延時間が増大し, クロック信号による同期に間に合わず誤作動を起こす故障を遅延故障と呼ぶ。遅延故障には, 立ち上がり (0 から 1 への変化) の遅延と, 立下り (1 から 0 への変化) の遅延の二通りの故障が考えられる。

また, 遅延故障は信号値の変化に起因するので, 故障を検出するためには, 変化前の信号値を設定するテストパターンと変化後の値を確かめるテストパターンの二つを連続して印加する必要がある, これを 2 パターンテストと呼ぶ。

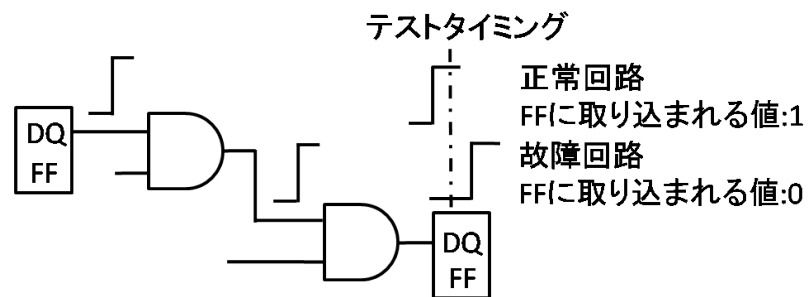


図 2.2. 遅延故障モデル

2.1.6. フィールドテスト

LSI が出荷後, 運用段階で行う検査をフィールドテストと呼ぶ. システムの起動時, 又は待機時にテストを行うため, 短時間で行う必要がある.

2.2 スキャンベース BIST [3]

スキャンベース BIST とは, テスト容易化設計の一つである. 図 2.3 にスキャンベース BIST の構成を示す.

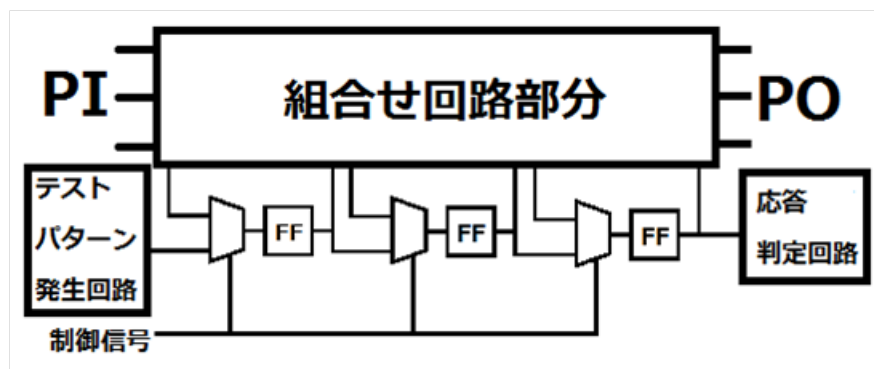


図 2.3. スキャンベース BIST の構成図

対象とする順序回路の FF をスキャンチェーンと呼ばれる直列につないだシフトレジスタに置き換え, スキャン出力まで値をシフトしていき内部状態の制御, 観測を行う. また, テストパターン発生回路とテスト応答判定回路を組込むことで簡易的な検査を可能にする. テストパターン発生回路には線形帰還シフトレジスタ (LFSR: Linear Feedback Shift Register) がよく用いられる. 本研究でも LFSR を使用した回路を想定している.

スキャンベース BIST はシフトモードとキャプチャモードを繰り返すことでテストを行う. シフトモードとは, スキャンインと呼ばれる LFSR が生成したテストパターンを各 FF に印加する動作と, スキャンアウトと呼ばれる結果の出力を行う動作の二つの動作を持つモードである. キャプチャモードとは, スキャンチェーンに印加したテストパターンを用いて回路の応答を FF にキャプチャするモードである. 一般的には一度のキャプチャモードにつき一回のキャプチャを行う. しかしながら, 対象とする故障モデルが遅延故障である場合は, 変化前の信号値と変化後の信号値を確かめる二つのテストパターンを連続で二回キャプチャを行う. これを, 2 パターンテスト [4] と呼ぶ.

LFSR が生成した疑似乱数をテストパターンとして利用するため, チップ上のメモリに記憶させるデータ量を削減できる. しかしながら, 検出に特定のテストパターンが必要な故障において, 疑似乱数を用いたテストパターンは有効でない. よって, 十分な故

障検出率を得るまでに膨大な時間を要するという欠点がある。この欠点は短時間でテストを行う必要のあるフィールドテストにおいて問題となる。この問題を解決する手法の一つとして次節で説明するマルチサイクルテストが存在する。

2.3 マルチサイクルテスト

マルチサイクルテストとは、スキャンベース BIST におけるテスト手法である。図 2.4 にマルチサイクルテストの基本概念を示す。

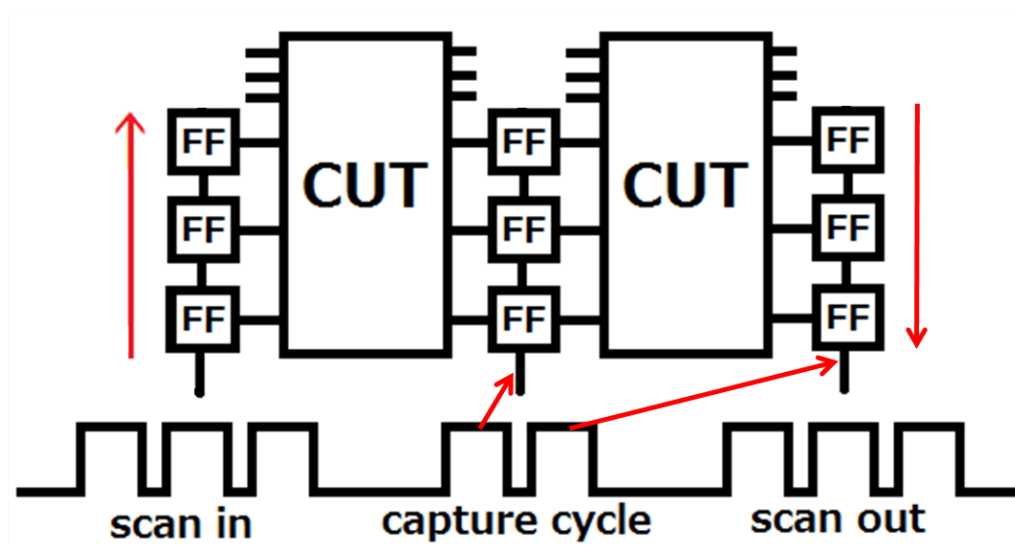


図 2.4. 時間展開したマルチサイクルテスト

マルチサイクルテストはキャプチャモード時に、連続でキャプチャを複数回行うテストである。キャプチャを連続で複数回行うことにより、キャプチャ途中の FF について検査可能となり、少ないテストパターンで多くの内部状態についてのテストができる。しかしながら、キャプチャモード時にキャプチャを複数回行うため、スキャンインからスキャンアウトまでの論理ゲートのパスが長くなってしまふ。そのため、テストパターンによって顕在化した故障の影響が、出力まで伝搬されない可能性がある。これは故障マスク現象と呼ばれ [5], 故障のマスクにより故障検出率が向上しない場合がある。

図 2.5 に故障マスク現象の例を示す。ゲート A の出力値に 1 縮退故障が生じているとする。一回目キャプチャではゲート A の正常値と故障値が異なるので、故障が励起しそ

の故障の影響がFFまで伝搬する。しかしながら、二回目キャプチャではNANDゲートに伝搬する故障値は出力値に影響を与えないので、故障の影響が二回目キャプチャで消失してしまう。

この故障マスクの問題は、全てのFFの値をキャプチャサイクル毎に観測するという、中間観測手法によって解決することができる。しかしながら、全てのFFを観測することは、新しくゲートや配線を施すという意味であり、回路の複雑化が生じ、信号の遅延、面積オーバーヘッドの増加などの新たな問題が生じてしまう。

この問題を解決する方法として、中間観測対象とするFFを選択して、一部に限定する部分観測手法が提案されている [5]。部分観測手法は、少ない観測点で多くの故障を検出できるような中間観測FFの選択が重要であるが、有効な選択手法は確立されていない。よって、本研究では、遅延故障の故障モデルに対して、縮退故障の故障モデルに用いられる中間観測FFの選択法を用いて実験を行い、遅延故障の故障モデルに対する中間観測マルチサイクルテストの効果を考察する。

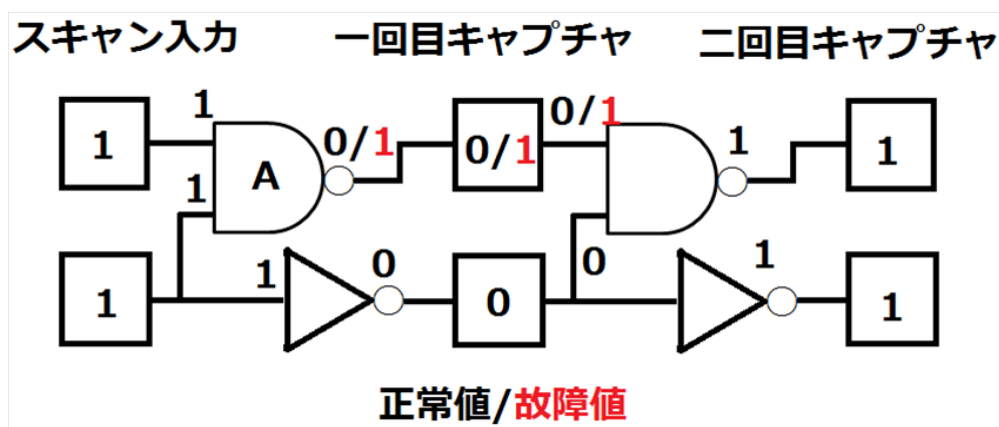


図 2.5. マスク故障の例

第 3 章

遅延故障の中間観測マルチサイクルテスト

本章では, 本論文の実験に用いる具体的な遅延故障のテスト方式と中間観測 FF の選択手法について説明する.

3.1 遅延故障のテスト方式

遅延故障とは, 同期式順序回路を構成する素子や信号線の信号変化の伝搬遅延時間が増加し, 次のクロック同期で取り込まれる際に, 遅延時間が決められた閾値を超えることで誤作動を生じる. また, 遅延故障には各故障個所に信号の立ち上がり (0 から 1 への変化) の遅延と, 立下り (1 から 0 への変化) の遅延の 2 通りの故障が考えられる.

スキャン設計がなされた順序回路では, テストパターンの印加とその応答の観測を, シフト動作で行う. 遅延故障は信号値の変化に起因するため, 故障の検出には, 変化前の信号値を設定するテストパターンと, 変化後の値を設定するテストパターンの二つを連続して印加しなければならない. この手法を 2 パターンテストと呼ぶ.

2 パターンテストで高い故障検出率を得るためには, テストパターンの印加方法を工夫しなければならない. スキャン動作は, 1 パターン目の印加と 2 パターン目の出力の観測に有効であるが, 1 パターン目と 2 パターン目を連続して実動作時間で印加することができないためである. この問題を解決する手法の一つとして, 次節で説明するラウンチオフキャプチャ方式 (LoC : Launch off Capture) が存在する.

3.2 LoC 方式

LOC 方式では, スキャンシフト動作で, 1 パターン目をスキャンインによりスキャンチェーンに設定した後, 通常動作で, システムクロックにより 2 パターン目 (ラUNCHサイクル) を設定し, 続けてキャプチャを行う. つまり, スキャンインをした後に, システムクロックにより二回キャプチャを行うことになる. 図 3.1 にクロック信号とスキャンイネーブル信号のタイミングチャートを示す.

LoC 方式の特徴として, ラUNCHサイクルが通常動作と同じなので設計時の制約が少ないことがあげられる. また, 正常な回路を不良と判定する過剰テストによる不用意な歩留まりの低下の危険性も少ない [6] こともあげられる.

本研究では, LoC 方式に対して図 3.2 に示すようにマルチサイクル化を施し, 次節で説明する中間観測 FF の選択法を用いて実験を行う.

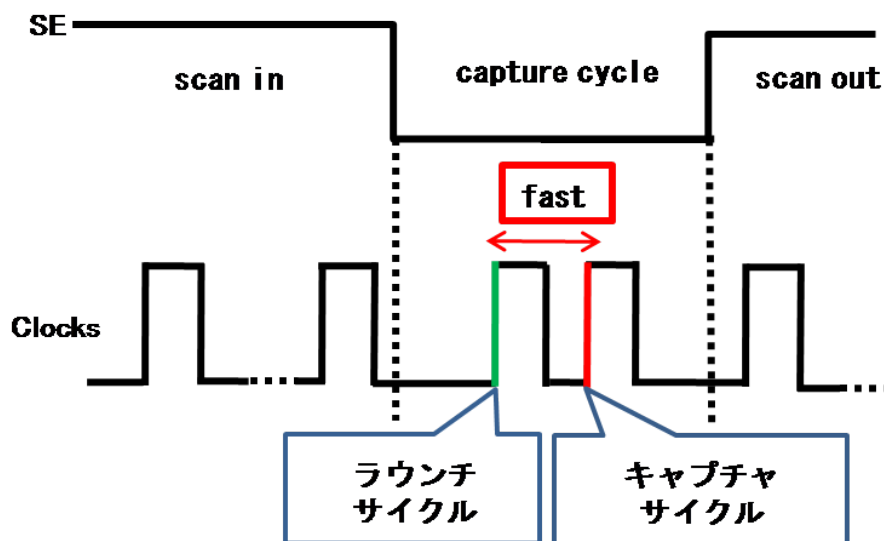


図 3.1. Launch off Capture のタイミングチャート

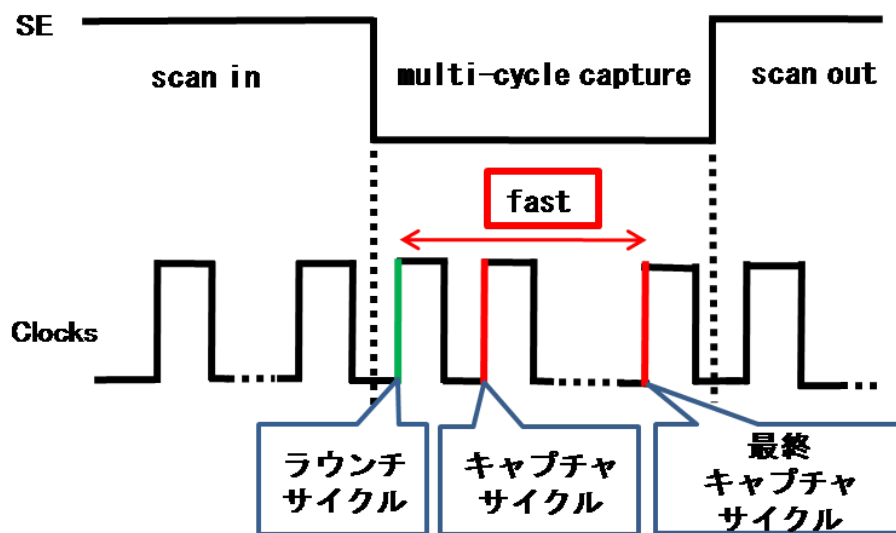


図 3.2. マルチサイクル Launch off Capture のタイミングチャート

3.3 回路構造の解析による中間観測 FF の選択手法

3.3.1. ゲート FF 接続複雑度による選択手法 (COMPLEX)

COMPLEX 手法はゲート FF の接続複雑度評価による選択法である。接続複雑度は、ゲートごとの論理的な接続情報 (入出力 FF 数, 入出力経路長, 経路上のゲート数) を解析し, FF から後方追跡可能なゲートの接続情報の合計値である。接続複雑度の高い FF では, より多くの故障を検出する可能性があると考えられており, 優先的に中間観測 FF として選択する。

各ゲートの出力に対して図 3.3 に示すような 6 つの評価値 P1~P6 を計算する。

P1 : 対象ゲートから到達可能な入力側 FF の総数

P2 : 対象ゲートから到達可能な出力側 FF の総数

P3 : 対象ゲートの入力側にある到達可能なゲート総数

P4 : 対象ゲートの出力側にある到達可能なゲート総数

P5 : 対象ゲートから入力側 FF までの最長経路長

P6 : 対象ゲートから出力側 FF までの最長経路長

その後, 求めた評価値を用いて, 以下の手順で FF の優先順位を決める。

Step1 : 評価値 P1~P6 をゲートごとに合計する。

Step2 : 各出力 FF に到達可能な経路上に存在するゲートの総評価値を合計してメトリックを求める。

Step3 : メトリックにより FF を順位付けする。

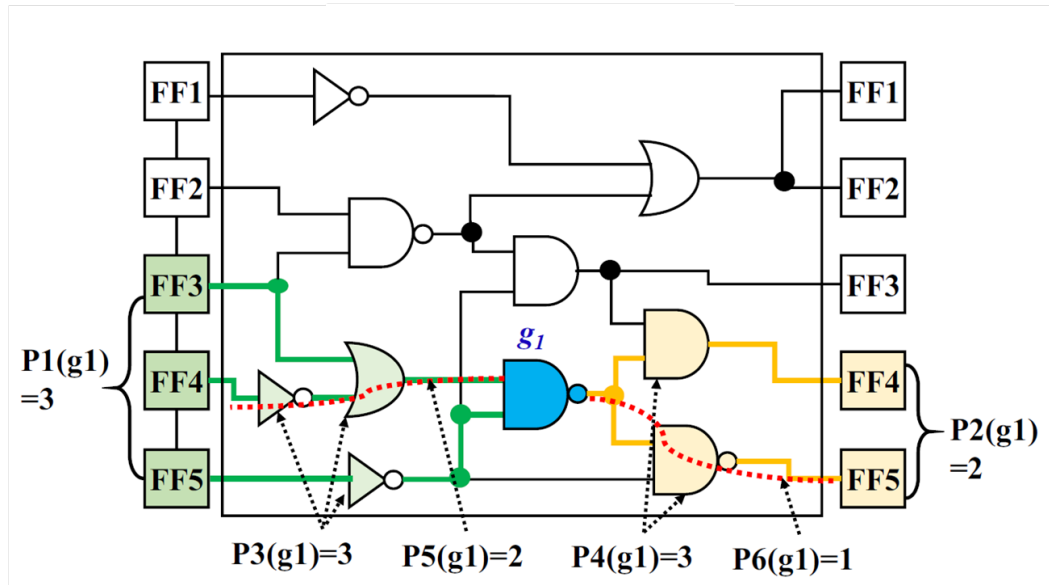


図 3.3. COMPLEX 手法の回路例

3.3.2. 時間展開を考慮した構造的な故障伝搬経路評価による選択手法

FF から後方追跡可能なゲートのうち, 異なる制約条件 (TYPE1~3) を満足するゲートの数を求める. 条件を満足するゲートの数が多い FF では, より多くの故障を検出する可能性があると考えられており, 優先的に中間観測 FF として選択する.

制約条件によって三つのタイプに分ける.

3.3.2.1. TYPE1

TYPE1. 各 FF に対して, 次の条件 1 および 2 を満足するゲート数を求める.

条件 1: 対象ゲートから到達可能な出力 FF は一つ (FFi) である.

条件 2: 対象ゲートの入力側の FF に FFi が含まれる.

TYPE1 のゲート例を図 3.4 に示す.

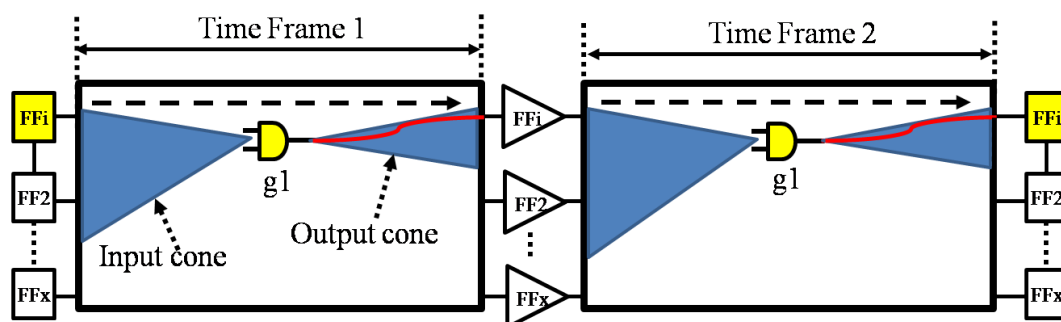


図 3.4. TYPE1 のゲート例

3.3.2.2. TYPE2

TYPE2. 各 FF に対して, 次の条件 1,2 および 3 を満足するゲート数を求める.

条件 1: 対象ゲートから到達可能な出力 FF は一つ (FFi) である.

条件 2: 対象ゲートの入力側の FF に FFi が含まれない.

条件 3: 入力側の FFi から出力側の FFi に到達可能 (経路がある) である.

TYPE2 のゲート例を図 3.5 に示す.

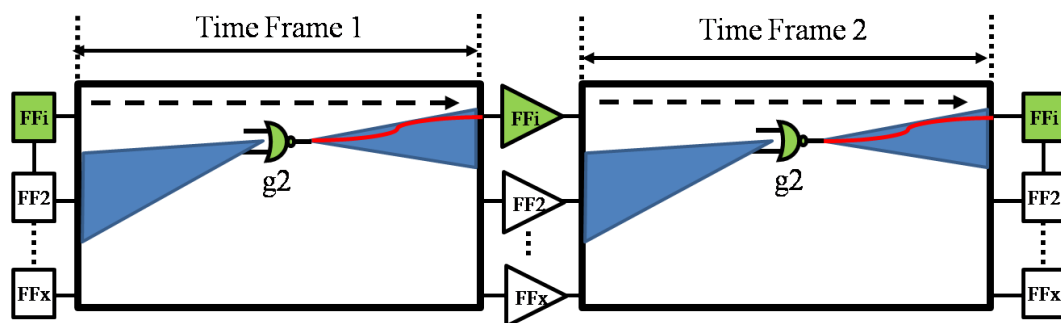


図 3.5. TYPE2 のゲート例

3.3.2.3. TYPE3

TYPE3. FF ごとに対して, 次の条件 1,2 および 3 を満足するゲート数を求める.

条件 1: 対象ゲートから到達可能な出力 FF は一つ (FFi) である.

条件 2: 対象ゲートの入力側の FF に FFi が含まれない.

条件 3: 入力側の FFi から出力側の FFi に到達不可能である.

TYPE3 のゲート例を図 3.6 に示す.

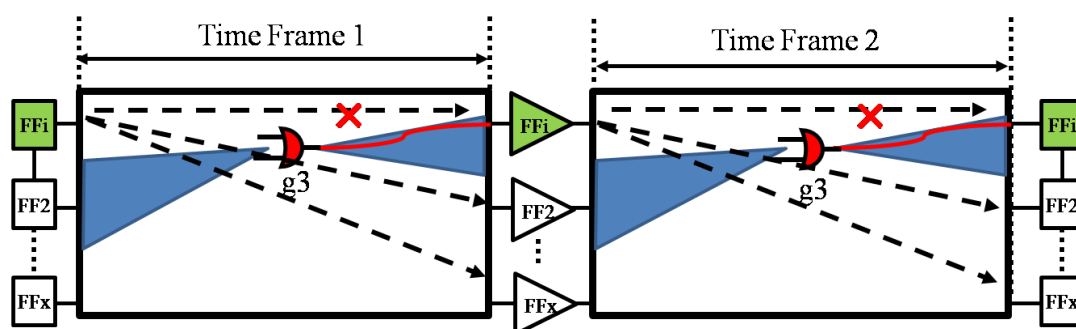


図 3.6. TYPE3 のゲート例

3.3.3. 分岐到達率評価による選択手法 (BRANCH)

FF から後方追跡可能な分岐数のうち, その FF まで到達可能な分岐の割合を分岐到達率と呼ぶ. 分岐到達率の高い FF ではより多くの故障を検出する可能性があると考えられており, 優先的に中間観測 FF として選択する. 分岐到達率評価の回路例を図 3.7 に示す.

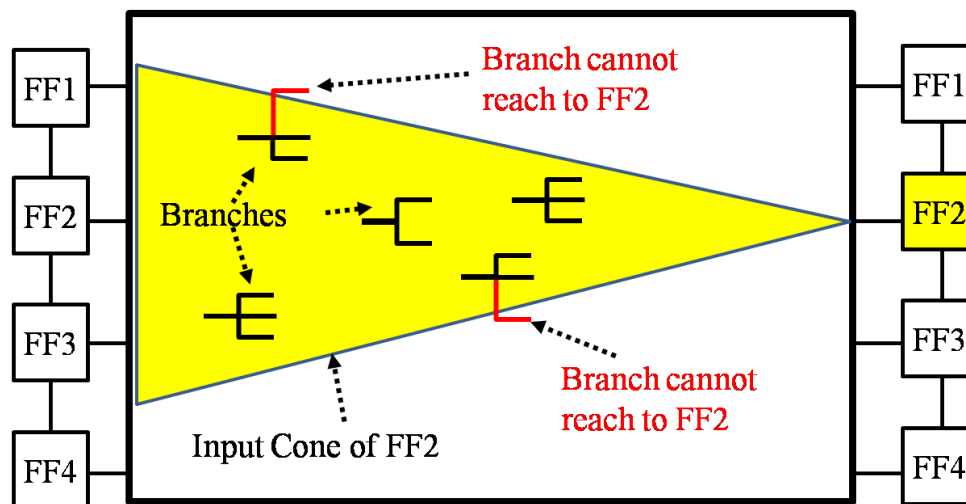


図 3.7. 分岐到達率評価の回路例

第 4 章

実験結果・考察

本章では, 本研究で行った実験の内容, 及び結果と考察について取り扱う.

4.1 実験環境

本実験では, 遅延故障検出に有効である LoC 方式に対してマルチサイクルテスト技術を組み込み, 中間観測 FF の選択法により, 遅延故障検出率がどのように変化するかを調べる. そのため, 選択した中間観測 FF を実際に被検査回路へ組み込み, プログラムを実行し, テスト数に対する故障検出率を算出した. また, 比較対象として, LoC 方式, 中間観測をもたないマルチサイクルテスト, 中間観測マルチサイクルテストにおいて全ての FF を観測する場合を同様に実験し, テスト数に対する故障検出率を算出した. 本実験では C 言語で作成したマルチサイクルテストのシミュレータを以下の表 4.1 に示す環境で実行した.

表4.1 計算機環境

CPU	Intel®core™2 Dou CPU E8600 3.33GHz
メモリ容量	8.00GB
OS	CentOS release 6.8

4.2 結果と考察

実験結果を図に示す。図は、テストパターン数に対して得られた遅延故障検出率のグラフと、各回路においての最大検出率と平均検出率を示す。

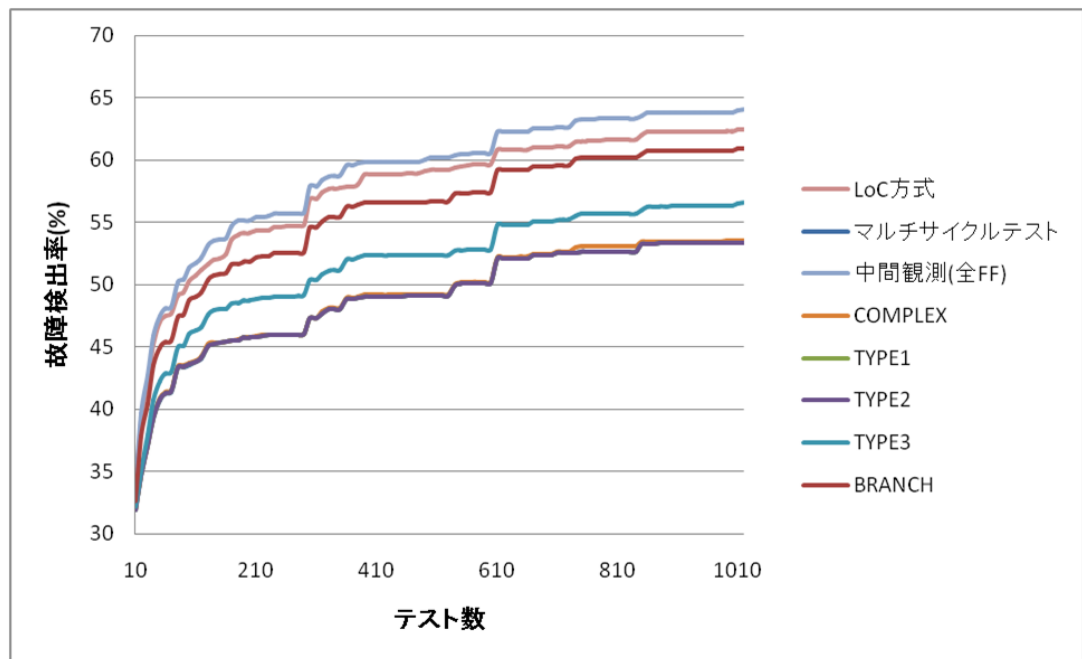


図 4.1. s5378 回路での故障検出率比較

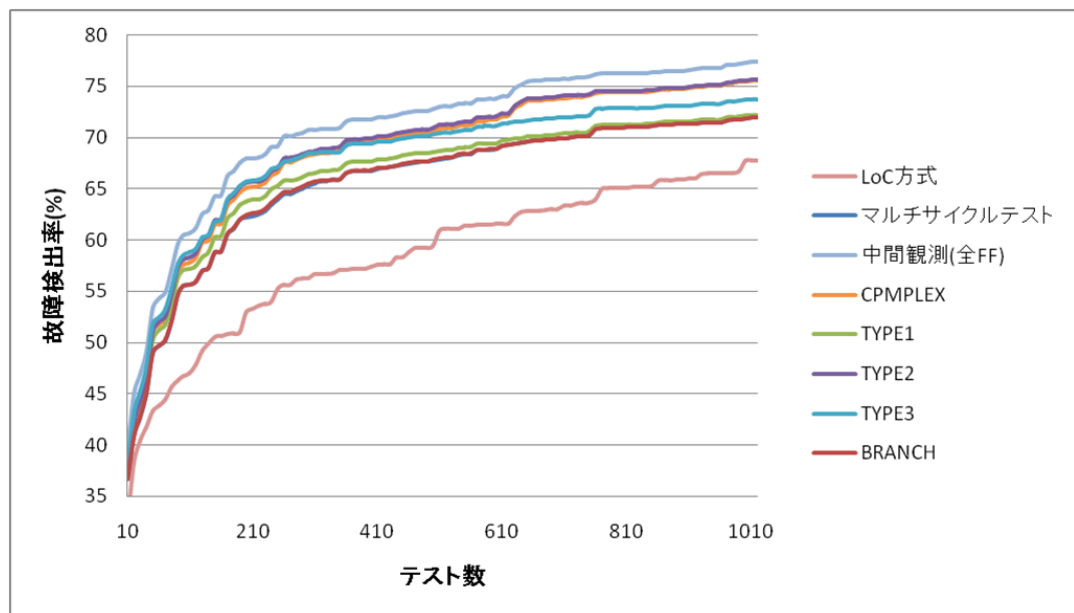


図 4.2. s9234 回路での故障検出率比較

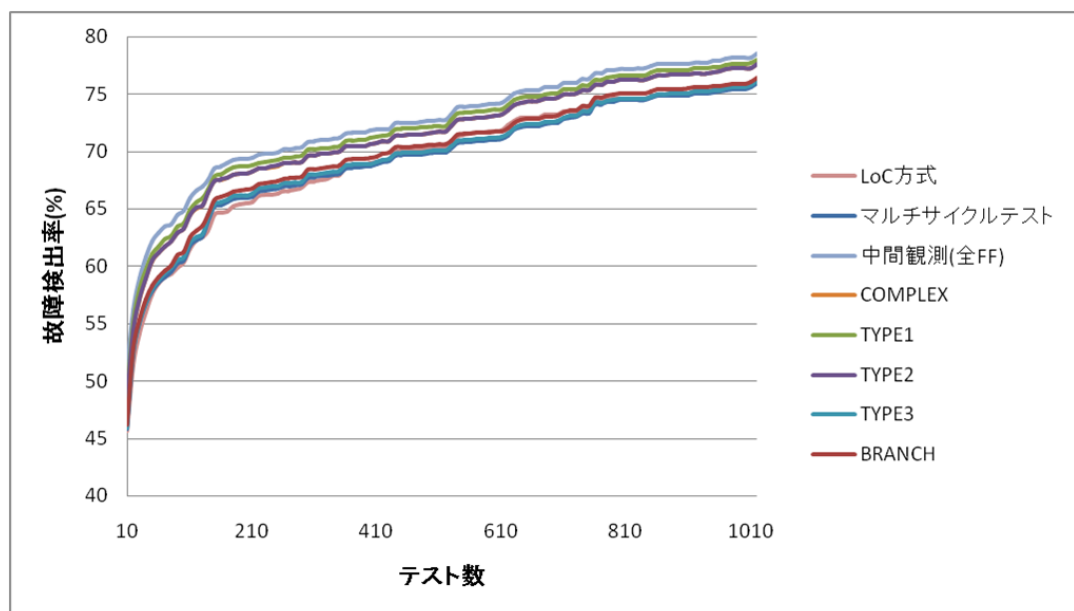


図 4.3. s15850 回路での故障検出率比較

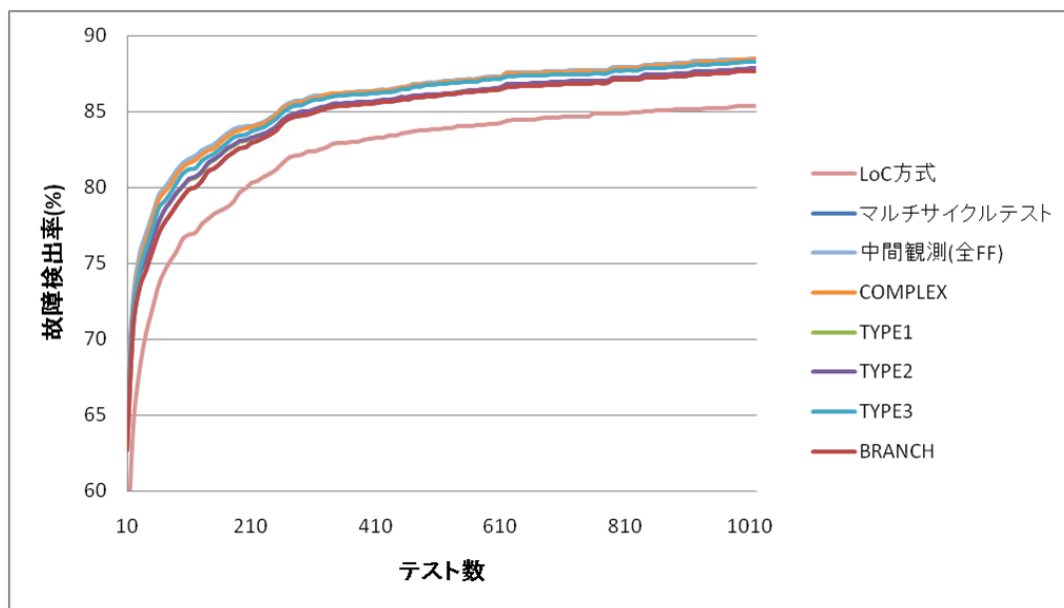


図 4.4. s38417 回路での故障検出率比較

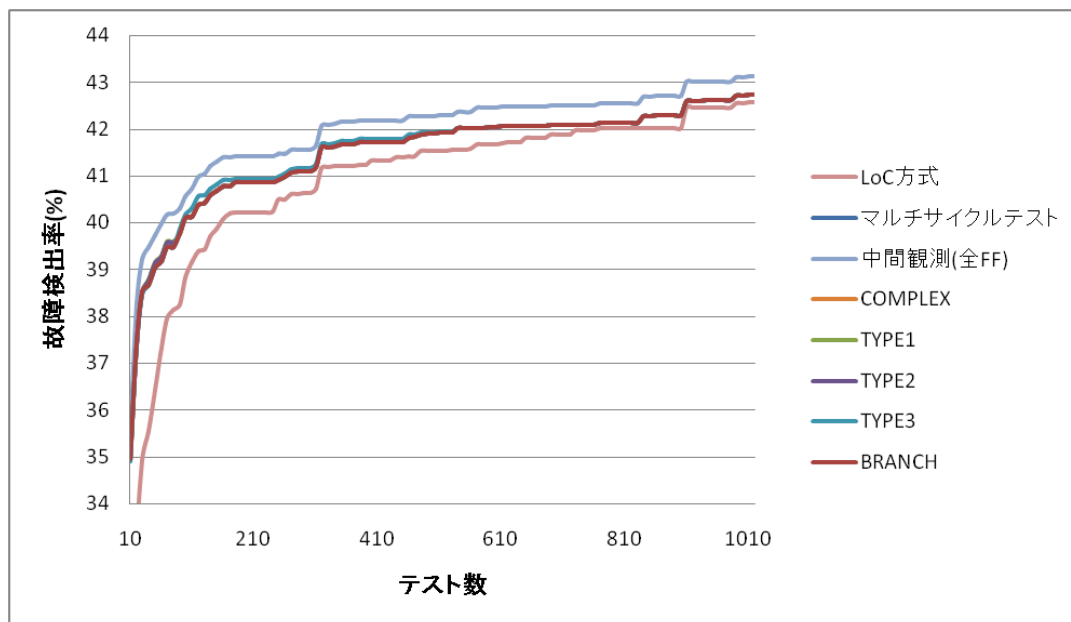


図 4.5. b05s.osaka 回路での故障検出率比較

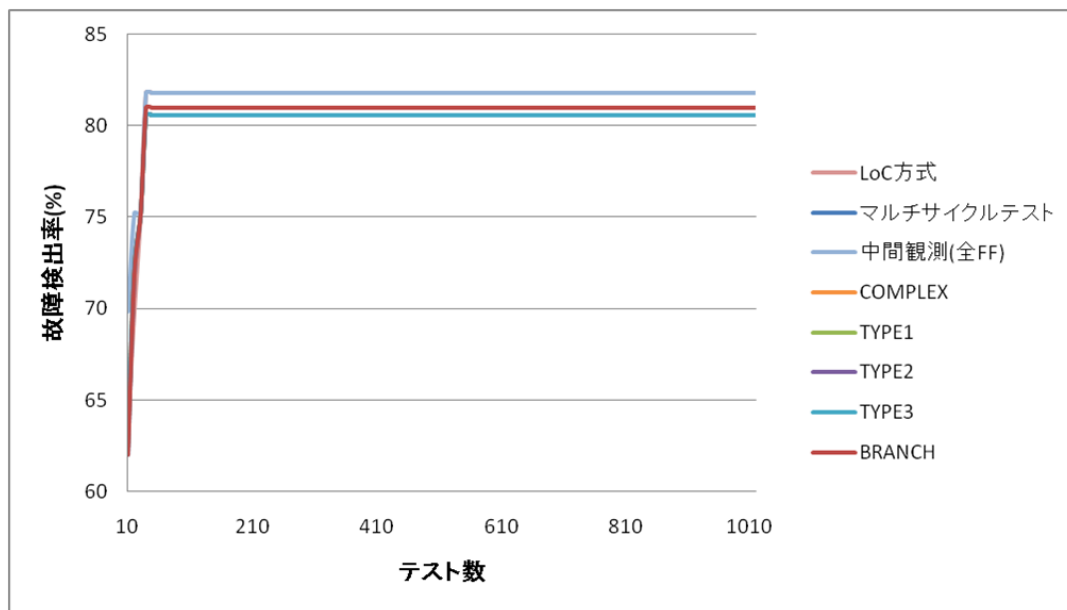


図 4.6. b06s.osaka 回路での故障検出率比較

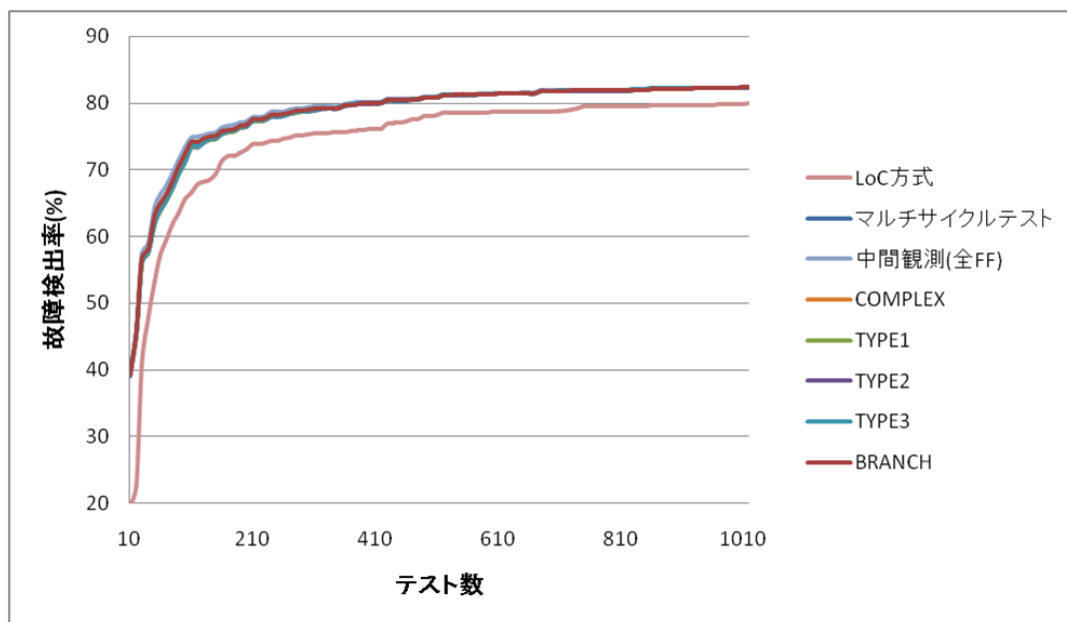


図 4.7. b14s.osaka 回路での故障検出率比較

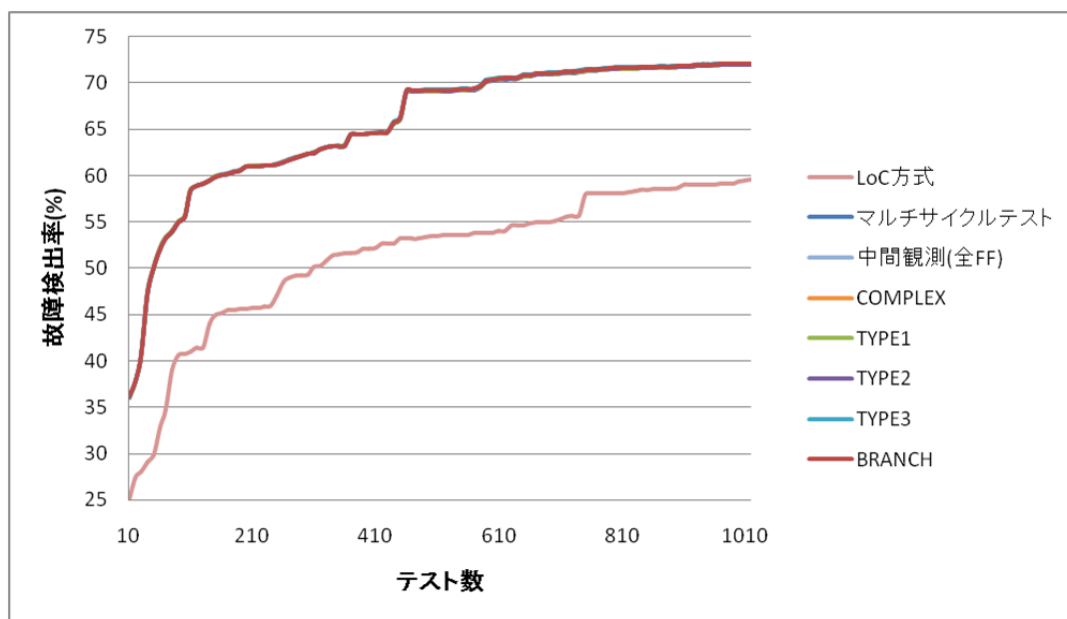


図 4.8. b15s.osaka 回路での故障検出率比較

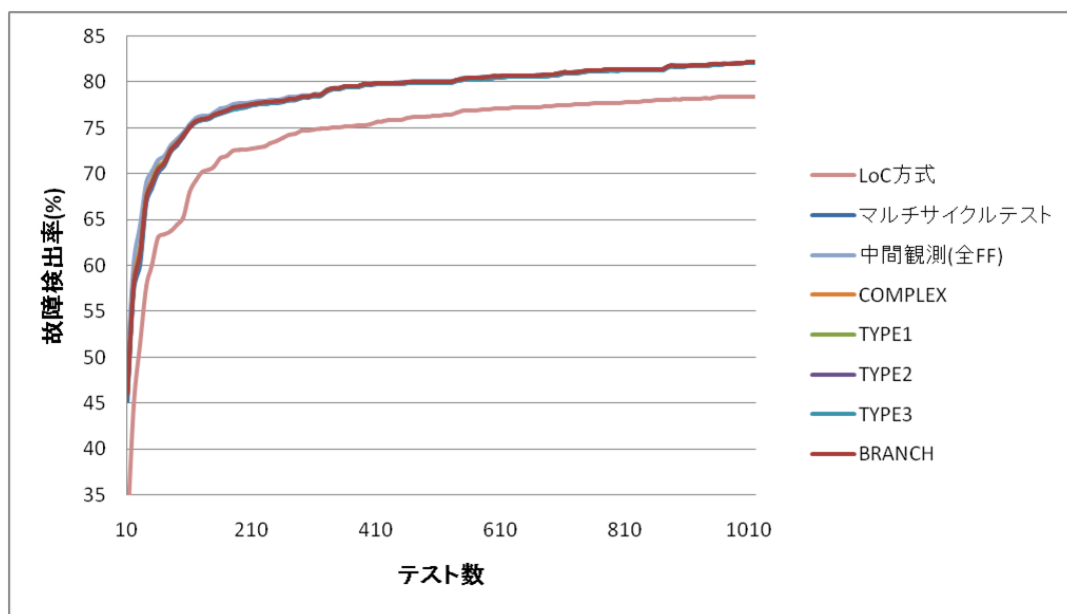


図 4.9. b20s.osaka 回路での故障検出率比較

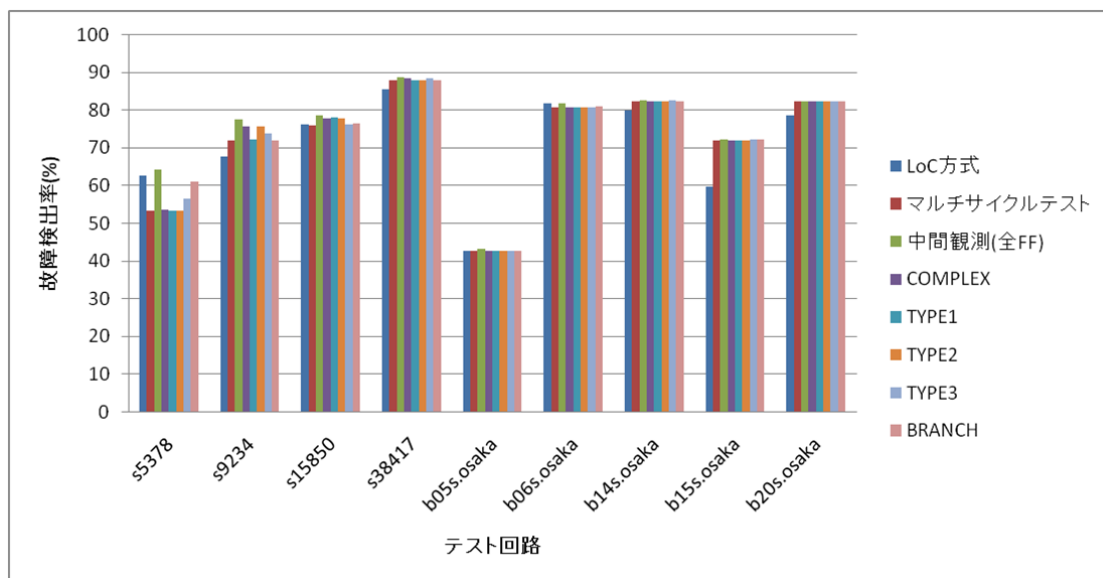


図 4.10. 各回路の最大検出率比較

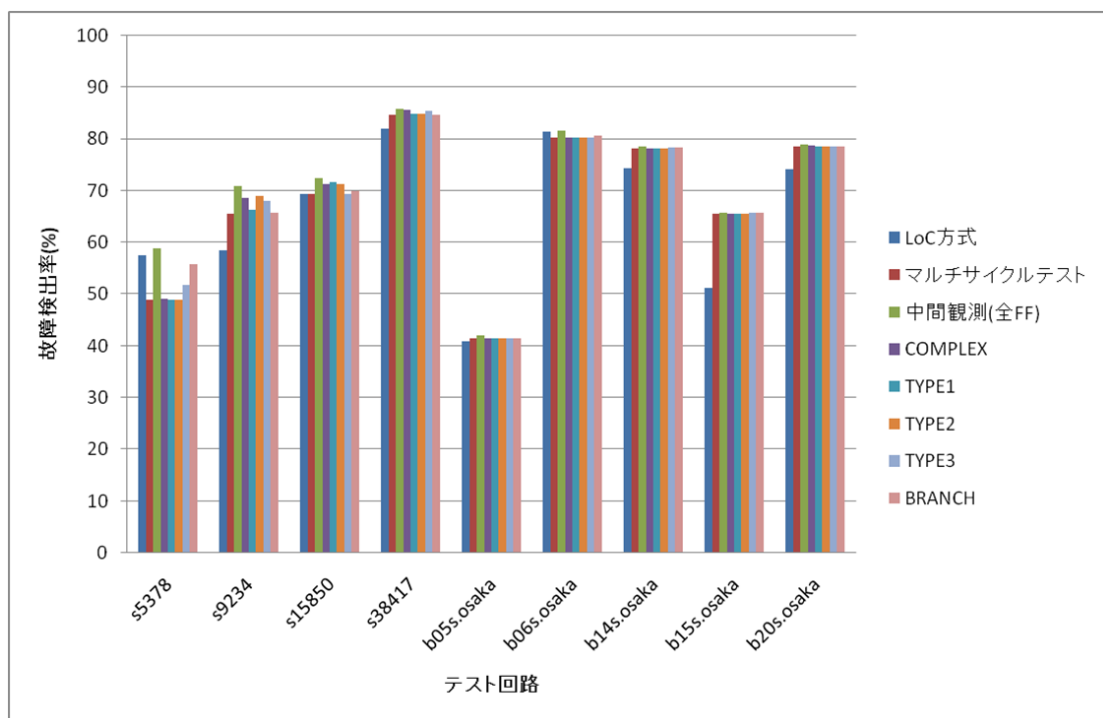


図 4.11. 各回路の平均検出率比較

LoC 方式と比較した場合,s5378 回路と b06s.osaka 回路において中間観測 FF の選択手法の検出率が LoC 方式の検出率を上回ることにはなかった。しかしながら,それら以外の回路では LoC 方式の検出率を上回る,良好な検出率であった。

中間観測を持たないマルチサイクルテストと比較した場合,全回路において中間観測 FF の選択手法の検出率がマルチサイクルテストの検出率を上回る結果となった。

中間観測 (全 FF) と比較した場合,全回路において中間観測 FF の選択手法の検出率が中間観測 (全 FF) の検出率を上回ることにはなかった。しかしながら,s38417 回路,b05s.osaka 回路,b14s.osaka 回路,b15s.osaka 回路,b20s.osaka 回路においては,中間観測 (全 FF) とほぼ同等な最大検出率を得られた。

中間観測 FF の選択手法を比較した場合,s5378 回路においては BRANCH と TYPE3 がよい検出率を得られた。s9234 回路においては,COMPLEX と TYPE2,次いで TYPE3 がよい検出率を得られた。s15850 回路においては,TYPE1,次いで COMPLEX と TYPE2 が比較的良い検出率を得られた。s38417 回路においては,COMPLEX と TYPE3 が比較的良い検出率を得られた。残りの回路においては,ほぼ同等な検出率を得られた。総合的に評価すると,COMPLEX と TYPE3 が全回路において比較的良い検出率を得られた。したがって,接続が多く,複雑な FF に着目して優先的に選択してやることが,故障検出率の向上が狙えるものであると考察する。

s5378 回路において,中間観測 (全 FF) 以外が LoC 方式の検出率を上回ることができなかった理由は,接続複雑度によって観測する FF を決定する中間観測 FF の選択法では,マスク故障を効率よく検出できなかったためであると考えられる。

第 5 章

あとがき

本研究では, スキャンベース BIST を用いた LoC 方式の中間観測マルチサイクルテストが遅延故障検出にどのような効果をもたらすのかを実験した. 実験は, すでに提案されている縮退故障を対象とした三つの中間観測 FF の選択法を用いて, 故障シミュレータによって遅延故障の検出率を求めた. また, 比較のために, LoC 方式, 中間観測をもたないマルチサイクルテスト, 中間観測マルチサイクルテストにおいて全ての FF を観測する場合の故障検出率も同様に算出した.

実験データの比較から, 遅延故障のマルチサイクル化は多くの場合で故障検出率の向上へ導いた. さらに, 縮退故障を対象とする三つの中間観測 FF の選択手法は, 遅延故障の中間観測マルチサイクルテストにおいても有効性を示した.

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, 審査頂いた本学岡野大准教授ならびに宇戸寿幸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情報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 (2010) 「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」
〈http://ci.nii.ac.jp/els/110008152343.pdf?id=ART0009665646&type=pdf&lang=jp&host=cinii&order_no=&ppv_type=0&lang_sw=&no=1485223013&cp=>〉
2017 年 1 月 24 日アクセス
- [2] 王 森レイ・門田一樹 (2016) 「中間観測 FF 順位付けアルゴリズム仕様書」 愛媛大学 高橋研究室
- [3] 藤原 秀雄 (1990) 『コンピュータの設計とテスト』 工学図書 p.221
- [4] 藤田 昌弘 (2006) 『システム LSI 設計工学』 オーム社 p.231
- [5] 濱田 宗・王 森レイ・樋上 喜信・高橋 寛 (2015) 「マルチサイクルテストにおける中間観測 FF の選択手法について」 第 72 回 FTC 研究会 熊本
- [6] 梶原 誠司・佐藤 康夫 (2011) 「論理回路に対する遅延テスト」
〈https://www.jstage.jst.go.jp/article/essfr/1/3/1_3_3_71/_pdf〉 2017 年 1 月 16 日アクセス.