

LSIの可検査性とグラフ構造における中心性の関係に関する考察 Consideration for the relationship between LSI testability and graph structure centrality

杉山 咲[†]
Sugiyama Saki

山口 賢一[†]
Yamaguchi Ken'ichi

岩田 大志[†]
Iwata Hiroshi

1. 研究背景

近年、半導体製造技術の向上によって、LSI(Large Scale Integration)の複雑化が進んでいる。それに伴い、LSIのテストが困難になりテスト費用も増加する問題がある。この問題を解消するため、テスト容易化設計(DFT: Design For Testability)という手法がとられている。

先行研究として、レジスタ転送レベルでの回路構造に着目し、特定の性質を回路に付与する DFT が提案されている。山口らはその性質を一般化した手法[1]を提案した。この提案手法では高い故障検出率を実現可能である一方で、設計変更に伴う面積増加や、スキャン設計に比べて設計変更が煩雑であるといった問題があった。

面積増加や設計変更コストを抑える方法の一つに、可検査性尺度を基準とする設計変更手法がある。可検査性とは、信号線に対するテストの容易さを数値化した指標である。外部入出力からの段数が多い場合やループがあるなど、内部状態の制御・観測が困難になるに応じて低くなる。可検査性が低い信号線を優先的に DFT の対象とすることで、設計変更の効率化やテスト費用の抑制が期待できる。しかし可検査性解析には回路動作の把握が必要となり、回路規模に伴う解析コストの増加が課題となっている。

そこで、グラフ構造の性質である中心性を応用することにより、回路内におけるテスト困難箇所の特定を効率化することを考える。中心性とは、各ノードがグラフ内においてどれだけ重要であるかを表す性質である。中心性を計算する際に考慮される要素は使用する指標によって変化するが、ノードの入次数やデータ流通量を考慮することができる。よって、ノードの中心性とモジュールの重要度には関係があると考えられる。そして、重要度の高いモジュールが持つ情報は多く、入力制御や出力パターンの解析コストが高くなる可能性が高い。つまり、重要度の高いモジュールに関連する信号線の可検査性も低くなると予想できる。

よって本稿では、中心性を基準として回路内における重要度の高いモジュールを検出し、該当モジュールに対しての設計変更によるテスト費用削減効果について調査する。また、回路構造をグラフ構造に変換し中心性を分析することで、可検査性の低い信号線が特定可能かを検証する。

2. 理論

2.1 テスト容易化設計(DFT: Design For Test)

テスト容易化設計とは、LSIを製造した際のテストを容易に行えるように回路を設計し、テスト費用の削減と故障検出率の向上を実現するための設計手法である。

LSIのテストは、任意のテストパターンを入力として与え、それに対する出力を観測することで回路内の故障有無を判断する。そこで回路が複雑な場合に内部モジュールの状態制御に制限が生じ、出力の観測が困難になるといった問題が生じる。

その問題への対策として、回路全体へのテストをより効率的に実施できるように、回路の設計を変更する。ただし DFT による設計変更では、変更後も本来の回路の挙動を保つ必要がある[2]。

現在提案されている DFT の一手法として、回路内にテストポイントを挿入する設計変更手法がある。テストポイントとは、内部の信号を可制御・可観測するために挿入される論理回路のことを指す。回路内の信号線にテストポイントを挿入することにより、信号線の挙動を外部から直接制御・観測することが可能となる。

2.2 可検査性

論理回路における可検査性は、回路に対しテストを生成・適用する際のコストに大きく影響を与える。可検査性が高いほど、その回路のテストが容易であることを表す[2]。テスト費用と可検査性には相関があり、回路内の可検査性が向上するとテストにかかる費用を少なくすることが可能となる。そのため、可検査性を向上させるための設計手法などがこれまでに多数提案されている。

そして、回路内で可検査性が低い箇所を把握するために回路の解析、該当箇所の可検査性の数量化を行うことを可検査性解析という[2]。可検査性解析を行うための尺度がこれまでに複数提案されてきているが、代表的なものとして、計算量(費用)を評価する尺度と、信号値 0, 1 の生起確率を評価する尺度がある。それぞれの尺度で、回路内の信号線の動作内容や結果を評価し、各信号線の可制御性や可観測性を計算する。可検査性解析を行うことで、設計変更基準制定やテスト生成アルゴリズムの効率向上の発見的手法の考案、故障シミュレーションの代替として故障検出率の評価などが行える。

2.3 故障検出率

故障検出率とは、想定した故障モデルのうち、テストパターンの印加によってどれだけの故障を検出できているかの割合を示し、テストパターン品質の評価指標として用いられる。故障検出率が高いほど、テストの質が高いといえる。故障検出率は式(1)によって求められる。

$$\text{故障検出率} = \frac{\text{テストによって検出できた故障数}}{\text{想定した全故障数}} \times 100[\%] \quad (1)$$

本稿では、設計前後の回路に故障シミュレーションを行い、故障検出率を比較する。故障シミュレーションでは故障回路と正常回路にテストパターンを印加した場合の回路

[†]奈良工業高等専門学校情報工学科. Department of Information Engineering, National Institute of Technology (KOSEN), Nara College.

動作シミュレーションを行い、出力値を比較することで検出できる故障を特定する。本稿では TetraMax を用いて故障シミュレーションを行う。TetraMax は入力したテストパターンに対する故障検出の可否のみを出力する。これより、提案するテスト系列の故障検出率の評価を行う [3]。

2.4 中心性

2.4.1 概要

中心性とは、ネットワーク構造におけるノードが持つ性質の一つである[4]。他のノードとの関連度合いや入力されるエッジの多さ、ネットワーク全体で考慮した際のデータの流通量など様々な観点からの指標が存在しており、ネットワーク構造の一要素としてネットワーク解析に用いられる。指標によって中心性の計算方法は異なる。

ネットワーク内において、該当ノードの中心性が高くなるほどネットワーク内での重要度が高いと考える。以下、本稿で用いる中心性について説明する。

2.4.2 固有ベクトル中心性

この中心性指標は、「中心的な頂点と隣接していると自身の中心性も大きい」という基準を基にしている。隣のノードの中心性が高ければ自身の中心性も向上し、同様に自身の中心性の大きさが周囲のノードの中心性にも影響を与える。自身の中心性については、ノードの入出次数を基準として計算する。本稿で用いるのは有効グラフであるため、ノードへの入次数をベースに中心性を計算する。

2.4.3 ページランク

この指標は、グーグルの検索エンジンにおいて Web ページのランク付けを行うためのアルゴリズムが基となっている。本指標では、以下の項目を基準としてノードの中心性が決定する。

1. 多くのページからリンクされるページは重要
2. 重要なページからリンクされるページは重要
3. 厳選されたリンクを受け取ることは貴重

これら 3 つの基準があることで、中心性の低いノードからのリンクが大量にある場合や中心性の高いノードから大量に他ノードに向けたリンクが飛んでいる場合など、極端な状況による影響を受けにくくなる。よって、純粋に重要度の高いノードの中心性が高くなる。

2.4.4 媒介中心性

媒介中心性は、ネットワーク上の流れが通りやすいノードの中心性が高くなる中心性である。ここでの「流れ」とは、ネットワーク上における各ノード間の最短経路のことを指す。つまり、流れが通りやすいノードは、グラフ内において重要度が高く、希少性の高い位置取りをしていると考えられる。

3. グラフの構築

本稿で使用したグラフの構築方法について説明する。

図 1 に回路構造とグラフ構造の対応を示す。RTL 回路上のモジュールにノード、信号線に対してエッジが対応する有向グラフとなっている。また、外部入出力に対してもノードを割り当てている。中心性の値を計算する際のノイズを減らすため、グラフ上ではデータパスのみを考慮し、制御信号は要素として考慮しない。つまり、回路上のデータパスがネットワーク上のフローに対応している。

なお、グラフの構築・可視化・中心性分析には Python パッケージである NetworkX[5]を使用した。実験に使用したベンチマーク回路は paulin, GCD, LWF の 3 つである。参考として、図 2 に paulin の回路図と変換後のネットワーク図を示す。

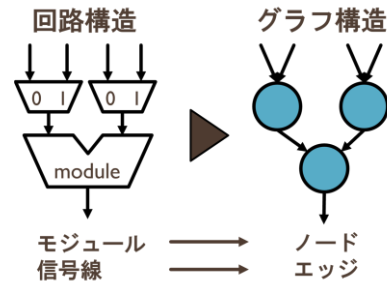


図 1 回路とグラフの構造対応

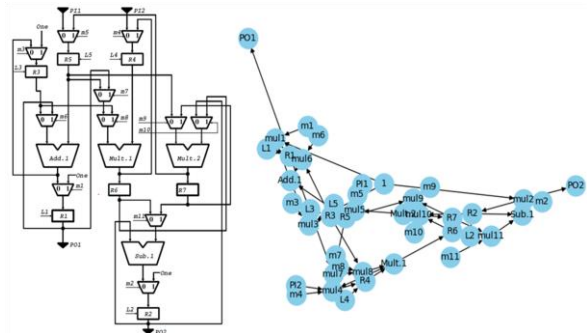


図 2 RTL 回路図と変換後のグラフ(paulin)

4. 実験

4.1.1 実験概要

本稿では、「回路構造に対する対応比較」と、「中心性の高いモジュールに対する DFT の効果検証」の二つの実験を行った。この実験を通して、中心性の指標ごとに回路構造との対応の程度に違いがみられるか、そして設計変更時の基準として有用であるかを検証した。

使用する中心性の指標により各ノードの中心性は大きく変動するため、モジュールの重要度に対応すると予想される中心性指標を選択し、回路構造との対応を調査した。今回使用した中心性指標は、固有ベクトル中心性、ページランク、媒介中心性の 3 指標である。使用した中心性指標の決定基準を説明する。固有ベクトル中心性、ページランクを選択した理由は入出力の複雑さと入出次数による中心性への影響に対応が見られると考えたためである。媒介中心性は、モジュールの状態数とネットワーク上の流れに対応が見られると考え選択した。

4.1.2 回路構造に対する対応比較

本実験では、構築したグラフに対して中心性を分析し、ノードが持つ中心性の高さとそれに対応する回路内モジュールの重要度に関係があるかを分析した。回路内モジュールの重要度については、ループや段数といった入出力の複

雑さ、信号の制御・観測の困難さを基にヒューリスティックにより判断した。

中心性指標によって素値の基準が異なるため、指標間での比較時には正規化した値を用いて比較している。また、本実験に使用した回路は paulin, GCD, LWF である。

4.1.3 中心性の高いモジュールに対する DFT の効果検証

本実験では最も中心性が高いノードに対応する回路内モジュールに対して設計変更を行った場合に、中心性指標ごとにテスト効率がどれだけ向上しているかを検証した。設計変更方法を図 3 に示す。設計変更対象モジュールの出力信号線に、テスト回路としてマルチプレクサを挿入する。

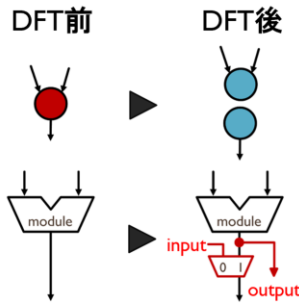


図 3 設計変更方法

設計変更によりモジュールの出力信号を外部出力として観測でき、外部入力から信号線に対する制御が可能となるため、該当モジュールの出力信号線の可検査性が向上する。

本実験では、中心性分析と設計変更、故障シミュレーションを 3 回繰り返した。その後、設計変更前の回路も含め故障シミュレーションの結果を比較し、故障検出率に変化が見られるかを調査した。

故障シミュレーションの実施条件について説明する。

故障シミュレーションは対象の RTL 回路に対して論理合成を行い、生成されたネットリストに対して実施した。使用したテストパターン数は 10,000 である。

既存回路の外部入力と設計変更により追加された入力端子に対し、32bit の LFSR を用いて乱数によるテストパターンを生成、入力・印加した。利用した LFSR の特性多項式は B97E8DC1₁₆ である。初期値は外部入力の数に応じて変化したため、表 1 に LFSR の初期値と外部入力数の対応を示す。

表 1 入力端子に接続する LFSR の初期値

外部入力数	初期値
2	D ₁₆ , F ₁₆
3	D ₁₆ , E ₁₆ , F ₁₆
4	C ₁₆ , D ₁₆ , E ₁₆ , F ₁₆
5	B ₁₆ , C ₁₆ , D ₁₆ , E ₁₆ , F ₁₆

また、マルチプレクサおよびレジスタの制御信号についても、LFSR により印加した。初期値は 3₁₆ とした。特性多項式については、設計変更回数により LFSR の bit 数が増加するため、それに伴って変動している。表 2 に LFSR の bit 数と特性多項式の対応を示す。

なお、本実験における対象回路は paulin のみとした。

表 2 制御端子に接続する LFSR の特性多項式

BIT 数	特性多項式
15BIT	4001 ₁₆
16BIT	8016 ₁₆
17BIT	10007 ₁₆
18BIT	20013 ₁₆

5. 実験結果

5.1.1 中心性指標ごとの回路構造に対する対応比較

ベンチマーク回路に対して中心性分析を行い、回路構造との対応がみられるかを分析した。図 4 に、paulin に対して実施した指標別の中心性分析結果を示す。図内の色分けは、モジュールの中心性を回路全体で正規化した値の区ごとに色を設定している。

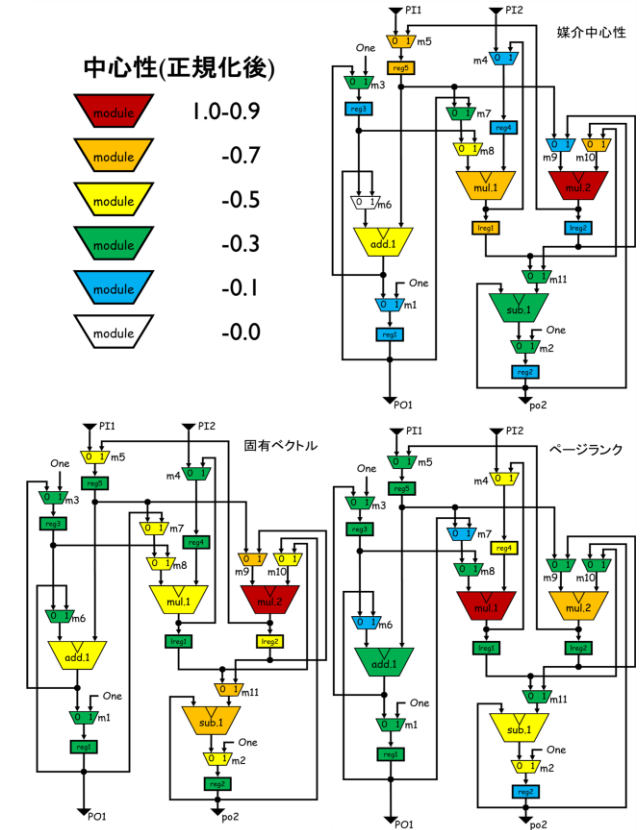


図 4 指標別の中心性分析結果(paulin)

結果として、固有ベクトル中心性と媒介中心性については対応が見られる可能性がある。入力にループが含まれているモジュールの中心性が高く、外部出力に対する段数が少ないモジュールの中心性が低い傾向がみられた。

ページランクも同様の傾向が見られたが、他の二つの指標よりも回路構造との対応は小さい結果となった。全体として値が中央に偏っており、モジュールの重要度の高さとの対応は弱いと考えられる。

5.1.2 中心性の高いモジュールに対する DFT の効果検証

固有ベクトル中心性、ページランク、媒介中心性それぞれに対して、中心性分析と設計変更を繰り返した。表 3 に、指標別に設計変更を行ったモジュールの一覧を示す。

設計変更に伴う中心性の値の変化と可検査性の値について述べる。3 指標の中で、可検査性に一番対応がみられたのは媒介中心性という結果になった。媒介中心性では、DFT の回数ごとに中心性の計算値差が全体的に小さくなっていった。また、DFT を行った想定のコモジュールに対応するノードの中心性がほぼ 0 になっており、可検査性指標の性質に近いと考えられる。固有ベクトル中心性とページランクについては、設計変更前後で中心性の値変動にあまり対応がみられなかった。

表 3 DFT を行ったモジュール

	1 回目	2 回目	3 回目
固有ベクトル	Mult2	mux10	Sub.1
ページランク	Mult1	Mux2	Mux8
媒介	Mult2	Mult1	add.1

故障シミュレーションを実施した結果について述べる。表 4 に、テストパターン数 10,000 時点での中心性指標と DFT 回数別故障検出率を示す。全体として、DFT 前から故障検出率が 2% 程度上昇していることがわかる。しかし、DFT の回数と故障検出率には対応がみられていない。全ての指標において、1 回目の設計変更時と比較して 3 回目の故障検出率は低下、もしくは変動が無い結果となった。

表 4 テストパターン数 10,000 時点の故障検出率

	DFT 前	1 回目	2 回目	3 回目
固有ベクトル	95.72%	97.95%	97.67%	97.90%
ページランク	95.72%	97.96%	97.67%	97.96%
媒介	95.72%	97.67%	97.96%	97.43%

テストパターン数に応じた故障検出率の上昇率について述べる。図 5 に、故障検出率の変化を可視化した対数グラフを示す。横軸がテストパターン数、縦軸が故障検出率である。また、凡例内の paulin は DFT 前の回路、他の凡例は DFT を実施したモジュール名である。

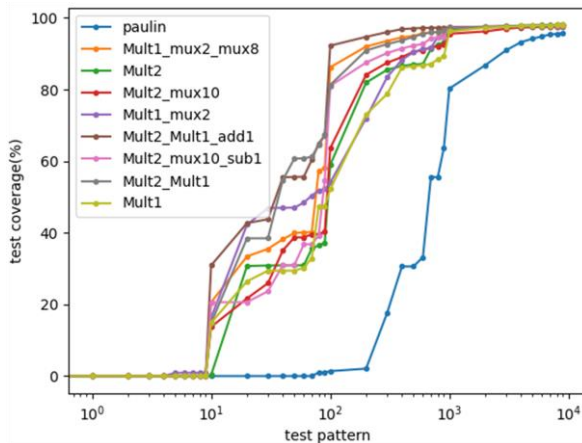


図 5 故障検出率の推移

図 5 から、DFT を行う前と比較して、DFT を行った場合の故障検出率の上昇速度が速いことがわかる。DFT を行っていない回路ではテストパターン数が 1000 を超えてから故障検出率が上昇しているが、DFT を行った場合にはテストパターン数が 100 の時点で大きく上昇している。

よって、DFT の基準として中心性指標は有用であり、テスト費用の削減にも効果的であると考えられる。

図 6 に、図 5 から DFT を 3 回行った際の故障検出率の推移のみを切り出したグラフを示す。

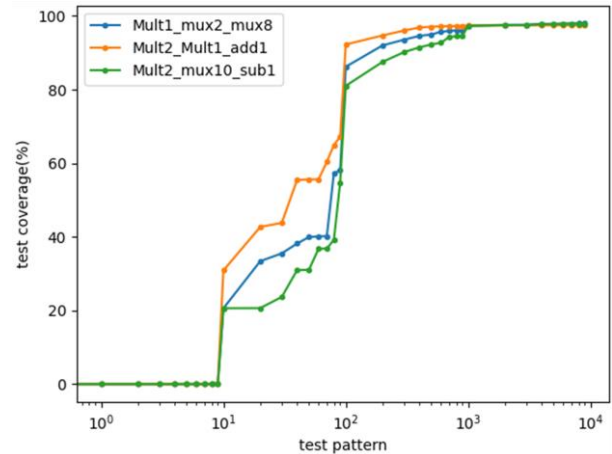


図 6 故障検出率の推移(DFT3 回目)

図 6 より、媒介中心性を基準として設計変更を行った場合が最もテストの効率が良いことがわかる。テストパターン数が 100 を超えてから 1000 程度までの故障検出率の増加が他二つに比べて大きい。媒介中心性、ページランク、固有ベクトル中心性の順に効率的であると言える。

6. おわりに

本稿では、回路内モジュールとネットワーク内の性質である中心性との対応を分析した。また、中心性を基準とした設計変更の効率化が行えるかについても検証した。

結果として、中心性指標は設計変更時の基準として有用であり、テスト費用削減にも効果が期待できることがわかった。特に媒介中心性については、可検査性と値の対応も見られた。

謝辞

本研究は、東京大学 VDEC 活動を通して、日本シノプシス合同会社の協力で行われたものである。

参考文献

- [1] Kenichi Yamaguchi, Michiko Inoue, Hideo Fujiwara, "Hierarchical BIST: Test-per-clock BIST with low overhead", ELECTRONICS AND COMMUNICATIONS IN JAPAN PART II-ELECTRONICS, Vol.90, No.6, pp47-58(2007).
- [2] Hideo Fujiwara, "Logic Testing and Design for Testability", pp133, (OCT, 1985).
- [3] 山口 賢一, 岩田 大志, "ATPG ツールを用いた組込み自己テストのための評価支援システムの構築", 奈良工業高等専門学校 研究紀要 43 号, pp.45-48, 2007.
- [4] 増田直紀, 今野紀雄, 複雑ネットワーク 基礎から応用まで. 近代科学社,(2010)
- [5] NetworkX developers, "NetworkX — NetworkX documentation", [https://networkx.org.\(2014-2024\)](https://networkx.org.(2014-2024))