

実数値シミュレーションに基づくテスト容易性評価モデル

伊達博 安浦寛人[†]

(財)九州システム情報技術研究所

〒814 福岡市早良区百道浜 2-1-22

[†]九州大学大学院システム情報科学研究科情報工学専攻

〒816 福岡県春日市春日公園 6-1

E-mail : date@k-isit.or.jp, yasura@c.csce.kyushu-u.ac.jp

あらまし 実数値シミュレーションは、テスト生成に応用され、高品質なテスト生成が可能であることが知られている。しかしながら、テスト容易化という観点からの議論は、十分とは言えない。

本論文では、論理ゲートレベルの回路に対して、実数値シミュレーションを用いたテスト容易性を判定するためのモデルを提案する。そして本モデルを用いたテスト容易化設計 DFHIT について述べ、今後の研究方針を述べる。

キーワード テスト容易化設計, VLSI, システム設計, 実数値シミュレーション, テスト生成

Toward a Testability Measurement Model Based on Real-Valued Logic Simulation

Hiroshi DATE and Hiroto YASUURA[†]

Institute of Systems & Information Technologies / KYUSHU

Momochihama 2-1-22, Sawara-ku, Fukuoka 814 Japan

[†] Department of Computer Science and Communication Engineering,

Graduate School of Information Science and Electrical Engineering, Kyushu University

Kasuga Kouen 6-1, Kasuga-shi, Fukuoka 816 Japan

E-mail : date@k-isit.or.jp, yasura@c.csce.kyushu-u.ac.jp

Abstract Real-valued logic simulation was applied to a test generation problem. Then it is reported to generate high quality test pattern. However from the view point of design for test, there is not sufficient discussion.

In this paper, We propose a model judging testability for logic gate level circuits. based on real-valued logic simulation, and describe a design for test using this model, called DFHIT. Also we describe our future plan.

key words Design for Testability, VLSI, System Design, Real-Valued Logic Simulation, Test Generation

1 はじめに

LSI 技術の向上とともに、携帯機器の普及が進み、組み込み型システムの需要が増大している。システム設計の鍵となるのは、効率良く目標性能を達成するための設計方式にかかっている。最終的にできあがるシステムの性能を初期の段階で評価できれば、高品質なシステム設計につながる。そのためには、性能評価モデルが必要である。システムの性能として評価しなければならない項目としては、テスト容易性、チップ面積、動作速度、消費電力、デバッグ/検証容易性、製品改良容易性、再利用性等があげられる。ここでは、これらの評価項目の中で、テスト容易性に注目する。LSI に対するテストの一つとして、テストパターンによる故障検出がある。テストパターンとは、故障が発生した場合と正常の場合とで、出力値の違いが生じるような入力パターン (0,1,不定値などの論理値の列) のことをいう。以下、特に断わらない限り、故障とは、信号線の値が、0 か 1 に固定する縮退故障を意味することとする。テスト容易性が高いとは、このテストパターンが作りやすいということの意味する。このテスト容易性を表す評価値が設計の初期の段階の情報で定義できれば、高品質な LSI 設計につながる。一般に論理 LSI は、組合せ回路部分と記憶素子をもつ順序回路であるため、時刻に依存して各ゲートにおける入出力値の変化を考慮しなければならない。よって、順序回路における故障を検出するための高品質なテストパターンを実用時間内で生成するには、自動テストパターン生成 (ATPG) とテスト容易化設計 (DFT) が重要な技術となる。

一方、論理シミュレーションを行なう場合に、論理値の代わりに実数値を用いることにより、与えられた入力パターンが対象とする故障のテストパターンに近いかどうかを判定することができることが報告されている [10]。

本論文では、この実数値を用いた論理シミュレーション (以下実数値シミュレーションと呼ぶ) の性質を利用したテスト容易性を評価するモデルを提案する。

以下では、2. で ATPG と DFT の現状について述べ、3. で実数値シミュレーションのテスト容易化設計への応用について述べる。4. で評価項目にまとめ、今後の方針を示す。

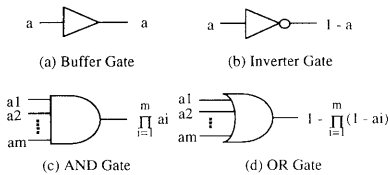


図 1: 実数値シミュレーションの例

2 テスト生成とテスト容易化設計の現状

順序回路を対象とした ATPG の研究としては、大きく分けて二種類ある。一つは、組合せ回路のテスト生成法である D アルゴリズム [14] や PODEM [8] を時間軸展開の概念を導入して拡張し、ある一つの故障に関して、それを検出するためのテストパターンを決定論的に構成して行く方法 (アルゴリズム方式) である。もう一つは、適当な初期パターンを与え、シミュレーションに基づき、それをテストパターンとなるように変形する方法 (シミュレーション方式) である [1, 3, 10]。

実数値シミュレーションに基づくテスト生成 [10] (以下、RSTG と呼ぶ) では、図 1 に示すように、論理回路における各ゲートの演算処理を実数値に拡張する。そしてある故障を検出するためのテストパターンと任意に与えられた入力パターンが、どの程度違うのかを表すコストを定義する。そして実数値の収束計算処理を繰り返すことにより、与えられた入力パターンをテストパターンへと近づけて行く。本論文では、この実数値の収束計算を用いたテスト容易化設計について論ずる。以下では、その準備として、RSTG の特徴とテスト容易化設計の現状について述べる。

2.1 RSTG の特徴

順序回路を L 時刻分だけ時間軸展開した回路において、検出すべき故障 f が与えられると、図 2 に示す手順で収束計算処理が行われる。最初に、収束計算処理の対象となる系列長 L の入力系列 ISP として、 ISO が選ばれ、 ISP に対して、正常時と故障時の実数値シミュレーションを実行する。その結果、故障 f が検出されなければ、以下の処理を繰り返す。入力系列 ISP において 1 ビットだけ値を反転させたものを新たな入力系列 ISA として選択する。そして、 ISA に対して、正常時と故障時の実数値シミュレーションを実行し、故障が検出されない場合は、各入力系列のコストを計算する。ここで、入力系列 IS のコスト $C(IS)$ とは、式 (1) で定義される。

$$C(IS) = \frac{1}{DO + DF} \quad (1)$$

DO は、正常時の実数値シミュレーションによる外部出力端子 i における時刻 k での出力値 $V_i(i, k, IS)$ と故障時の実数値シミュレーションによる外部出力端子 i における時刻 k での出力値 $V_f(i, k, IS)$ の差を全時刻 (L : 時間軸展開数) 及び全外部出力端子 (M : 全外部出力端子数) について加えたものである。

$$DO = \sum_{k=1}^L \sum_{i=1}^M |V_i(i, k, IS) - V_f(i, k, IS)| \quad (2)$$

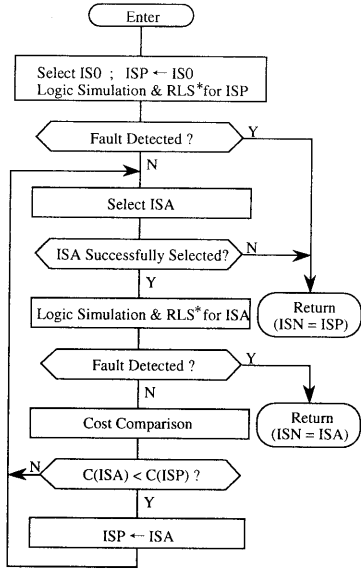


図 2: 収束計算処理

DFは、正常時の実数値シミュレーションによる記憶素子 j における時刻 L での出力値 $S_i(j, IS)$ と故障時の実数値シミュレーションによる記憶素子 j における時刻 L での出力値 $S_f(j, IS)$ の差に重み W_j をかけたものを全ての記憶素子 N (全記憶素子数) について加えたものである。この重み W_j を制御することで故障信号を外部出力に近い記憶素子へと誘導している。

$$DF = \sum_{j=1}^N W_j |S_i(j, IS) - S_f(j, IS)|. \quad (3)$$

順序回路テスト生成の処理手順を図3に示す。未検出故障が選択されると収束計算処理により検出されたテストパターン ISN をテストパターン系列に加える。その後、故障シミュレーションを行い、同時に検出される故障を対象故障から除く。回路状態を記憶し、シミュレーションの結果から過去に同じ回路状態があったかを調べることで、故障検出における不要な処理を回避している。

文献 [4] では、RSTG とアルゴリズム方式 ATPG に対して、故障検出率等の比較をしており、両方式を相補的に用いることが有効であると示唆している。RSTG の処理効率を向上させるようなテスト容易化設計を行なうことにより、テスト設計の品質向上が期待される。

実数値シミュレーションの特徴をまとめると以下のようになる。

- (1) 冗長故障の判定が出来ない [10].
- (2) アルゴリズム方式 ATPG と組み合わせると効果的 [4].

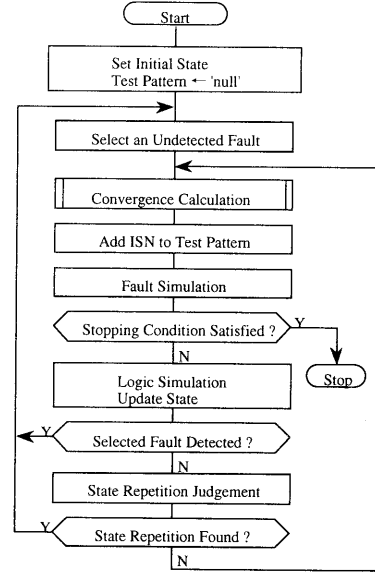


図 3: 順序回路テスト生成処理概要

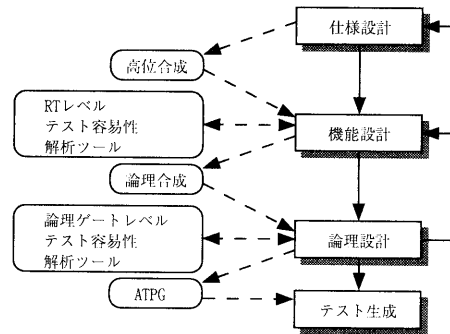


図 4: テスト容易化設計処理概要

(3) 並列処理による高速化が可能である [4].

2.2 テスト容易化設計の現状

図4にテスト容易化設計の処理概要を示す。一般に用いられているDFTとして、フルスキャン設計 [6] がある。この手法は、回路中の全てのフリップフロップ (FF) を観測/制御可能なスキャンFFとするため、順序回路のATPGを組合せ回路のATPGに簡単化できる。しかしながら、テスト回路面積の増大等の問題があるため、一部のFFをスキャンFFとするパーシャルスキャン設計 [12] が提案された。従来、これらのDFTは、論理設計後に行なわれていたが、論理設計とテスト設計を切り離すことが出来るという利点を持つ半面、スキャンFF挿入によるチップ面積の

増大や、タイミングの再設計による工数増等の問題が発生する。

そこで、これらの問題に対処するため、機能レベル (RT レベル) または、より高位のレベルで得られる回路情報をテスト容易化設計に利用し、論理ゲートレベルの回路の品質を向上させる研究が進められている。

RT レベルでの DFT としては、ゲートレベルでのテスト容易性評価法である SCOPE[9] を拡張し、スキャン FF の挿入点を決める手法 [13]、RT レベルで定義したモデル上でのフィードバックループに着目し、スキャン FF やマルチプレクサの追加点を決定する手法 [5]、演算器にスルー演算をもたせることによる DFT[16]、等が提案されている。RT レベルで、スキャン FF をもちいないノンスキャン DFT としては、マルチプレクサを用いて、テスト可能な回路構造を持つように回路分割をおこない、各部分回路の入力を選択するためのコントローラを付加することにより、フルスキャン設計と比較し、テスト回路面積とテストパターン数を削減する方式が提案されている [11]。RT レベルモデルで転送の単位となる“語”を解析の単位とし、語が取り得る値である“データ量”の RT レベルモデル上での伝搬解析を行なう事によりテスト容易性解析をするもの [15] も提案されている。

また、機能レベルより高位の情報を用いた DFT としては、動作レベルの情報を制御データフロログラフを用いて表現し、RT レベルの回路を合成する際に段階的に DFT を行なう手法 [7] が提案されている。

アルゴリズム方式 ATPG の場合は、仮定した故障に対して、観測ピンで故障が検出できるように入力パターンを決定論的に求める。よって、本当にその故障が設定できるのか (制御性)、そして故障が観測ピンまで、伝搬するのかどうか (観測性) が判ると、ATPG の負荷を軽減できる。そのため、従来のテスト容易化設計では、制御性、観測性を如何にして、設計の初期段階で評価するかを目的として研究が進められてきた。

このように従来の DFT は、アルゴリズム方式 ATPG を実行する際の負荷をなるべく小さくすることを目的としているため、シミュレーション方式のテスト生成を行なう場合には、回路のテスト容易性を正確に評価しているとはいえない。

3 実数値シミュレーションに基づくテスト容易性評価モデル

実数値シミュレーションをテスト容易化設計に利用する場合のアプローチの方法として、次の二つが考えられる。

一つは、ATPG を効率的に行なうための解析ツールとして実数値シミュレーションを用いる方法、そしてもう一つは、ATPG として、実数値シミュレーションを用い、それ

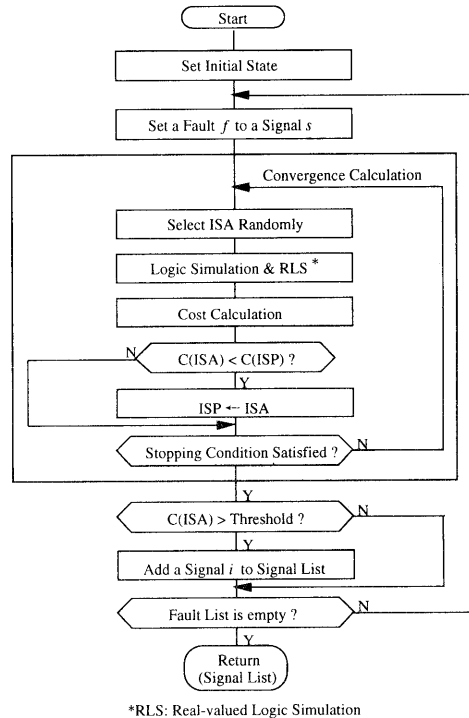


図 5: TIRS 方式の概要

を効率的に行なうためのテスト容易化設計を模索する方法である。以下では、それぞれのアプローチ方法について述べ、これらに基づくテスト容易化設計について述べる。

3.1 ATPG 効率化のための実数値シミュレーション

実数値シミュレーションを用いたスキャン FF やマルチプレクサなどの観測点を追加する TIRS (Test Insertion Using Real-Valued Logic Simulation) 方式を提案する。その概要を図 5 に示す。論理回路の各信号線に対して、故障を設定し、ランダムに入力パターンを生成し、実数値に置き換える。次に収束計算処理 (図 2 参照) を行なう。与えられた回数だけ上記の処理を繰り返す。このとき、収束計算処理によって、式 (1) で示すコストを計算し、改善されないと判断された場合は、その信号線を観測点の候補として登録する。全ての信号線に対してこれらの処理を行ない、得られた観測点の候補に対して、従来の観測点の決定方式を適用する。

3.2 RSTG を考慮したテスト容易化設計

ATPG として RSTG を採用した場合、テスト生成のコストを軽減するための方策としては、冗長故障の除去が考

えられる。

RSTG は、冗長故障が判定できないという特徴をもっているため、前もって冗長故障が除去されていると、テスト生成コストが削減できる。しかしながら、冗長故障を判定するには、テスト生成を行なうのと同じ計算量が必要なので、文献 [4] では、簡易的な冗長故障判定手法を用いた RSTG の効率化を行なっている。

3.3 テスト容易化設計 DFHIT

アルゴリズム方式 ATPG と RSTG とを組み合わせた ATPG を対象としたテスト容易化設計 DFHIT(Design for Hibrid Automatic Test Pattern Generation) について述べる。

ATPG として、アルゴリズム方式 ATPG と RSTG を組み合わせると、効率化が図れることは、文献 [4] でも指摘されている。この場合、DFT としては、従来の RT レベルおよび、論理ゲートレベルの DFT に加え、上記のテスト容易化設計を行なうことにより更に効率化が図れると考えられる。

4 今後の方針

4.1 評価項目

今後の LSI の大規模化に対処するため、高速化、高品質化という観点から以下の評価項目を設定する。

- (1) TIRS 方式の精度に関する評価
- (2) テスト容易化設計 DFHIT の効果に関する評価
- (3) 並列処理による高速化に関する評価

4.2 今後の方針

今後の方針としては、以下の点があげられる。

- (1) TIRS 方式の RT レベルへの拡張
- (2) 本モデルに基づくツールの実装と評価
- (3) システム設計支援ツールとしての統合

5 むすび

実数値シミュレーションをテスト容易化設計に利用する上での課題を明確にした。また、実数値シミュレーションに基づくテスト容易性評価モデルについて考察し、ATPG 効率化のための実数値シミュレーションに基づく観測点追加方式 TIRS を提案した。

さらに、アルゴリズム方式 ATPG と実数値シミュレーションに基づく ATPG とを組み合わせた、テスト容易化設計 DFHIT にも言及した。今後は、本システムを実装し、実際の LSI 設計に用いることにより、有効性を確かめたい。

参考文献

- [1] V. D. Agrawal, K.- T. Cheng and P. Agrawal, "CONTEST : A Concurrent Test Generator for Sequential Circuits," *Proc. 25th DA Conf.*, pp. 84-89, 1988.
- [2] F. Brglez, D. Bryan and K. Kozuminski, "Combinational Profiles of Sequential Benchmark Circuits," *Proc. ISCAS'89*, pp. 1929-1934, 1989.
- [3] K.- T. Cheng and V. D. Agrawal, "A Sequential Circuit Test Generator Using Threshold-Value Simulation," *Dig. FTCS-18*, pp. 24-29, 1988.
- [4] 伊達博, 中尾教伸, 畠山一実, "実数値シミュレーションに基づく並列順序回路テスト生成システム DESCARTES", *信学論 (A)*, Vol. J79-A, no. 1, pp. 47-56, 1996.
- [5] S. Dey and M. Potkonjak, "Non-Scan Design-For-Testability of RT-Level Data Paths," *Proc. IEEE ICCAD*, pp. 640-645, 1994.
- [6] E. B. Eichelberger and T.W. Williams, "A Logic Design Structure for LSI Testability" *Proc. 14th DA Conf.*, pp. 462-468, 1977.
- [7] I. Ghosh, A. Raghunathan and N. k. Jha, "Design for Hierarchical Testability of RTL Circuits Obtained by Behavioral Synthesis," *Proc. IEEE IC-CAD*, pp. 173-179, 1995.
- [8] P. Goel, "An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits," *IEEE Trans. Comput.*, Vol. C-30, No. 3, pp. 215-222, 1981.
- [9] L. H. Goldstein, "Controllability/Observability Analysis of Digital Circuits," *IEEE Trans. Circuit and Systems*, Vol. CAS-26, pp. 685-693, 1979.
- [10] 彦根和文, 池田光二, 畠山一実, 林照峯, "実数値シミュレーションを利用した順序回路テスト生成," *信学論 (D-I)*, Vol. J75-D-I, no. 11, pp. 1089-1098, 1992.
- [11] 細川利典, 川口謙一, 太田光保, 村岡道明, "RTL 回路分割を用いたテスト容易化設計手法," *DA シンポジウム'96*, pp. 225-230, 1996.
- [12] H-K. T. Ma, S. Devada, A. R. Newton and A. Sangiovanni-Vincentelli, "An Incomplete Scan Design Approach to Test Generation for Sequential Machine," *Proc. ITC'88*, pp. 730-734, 1988.
- [13] A. Motohara, S. Takeoka, T. Hosokawa, M. Ohta, Y. Takai, M. Matsumoto and M. Muraoka, "Design for Testability Using Register-Transfer Level Partial Scan Selection," *Proc. ASPDAC95*, pp. 209-215, 1995.
- [14] J. P. Roth, W. G. Bouricious and P. R. Schneider, "Programmed Algorithms to Compute Tests to Detect and Distinguish between Failures in Logic Circuits," *IEEE Trans. Electron. Comput.*, Vol. EC-16, No. 5, pp. 567-579, 1967.
- [15] 桜井涼二, 野田浩明, 高橋瑞樹, 神戸尚志, "RT レベルテスト容易性評価システム," *DA シンポジウム'96*, pp. 219-224, 1996.
- [16] 高島勝之, 井上美智子, 増澤利光, 藤原秀雄, "スルー演算を用いた非スキャン方式によるデータパスのテスト容易化設計," *信学技報 VLD*, pp. 15-22, 1996.