

## コントローラの遷移故障検出率向上のための コントローラ拡大法

日大生産工(院) ○飯塚 恭平

日大生産工 細川 利典

日大生産工 山崎 紘史

京産大 吉村正義

### 1. はじめに

近年、大規模集積回路(Large Scale Integrated Circuits : LSI)の微細化や高速化、電源電圧の低下に伴い、遷移故障モデル[1]などのタイミング欠陥に対するテストが必要とされ、遷移故障モデルにおいて高い故障検出率が求められている[2]。製造テストにおいて、遷移故障を検出できない場合、潜在的な故障を見逃す恐れがある。さらに、遷移故障に関するテストは、LSIの経年劣化による故障を検知するため、製造テストだけでなくオンラインテストに用いられている。

遷移故障モデルの故障検出率の向上の手法として、テスト容易化設計手法(Design for Testability: DFT)の一つであるスキャン設計[3]が広く用いられている。スキャン設計では、回路中のFlip Flop(FF)がシフトレジスタ状に接続される。シフトイン動作を用いてFFに値を設定し、2クロック通常動作をさせ、通常動作時の応答をシフトアウト動作で観測するテスト容易化設計である。

スキャン設計されたLSIに対して、タイミング欠陥に対する実速度スキャンテスト方式として、ブロードサイド方式[4]が提案されている。この方式は、LSIに連続した2つのテストを高速に印加させる2パタンテスト方式であり、スキャン設計されたLSIでは2パタンテストが実現可能である。

ブロードサイド方式の遷移故障のテスト生成モデルにおいて、故障を検出するテストパターンを生成するための必要条件は、故障の初期値を設定する状態と故障を励起し、その影響をFFに伝搬する2つの状態が存在し、これら2つの状態間の状態遷移が存在することである。レジスタ転送レベル(Register Transfer Level :RTL)の回路の設計時に、どのような状態遷移をするかが決定される。この状態遷移の設計によって、テスト不能故障と分類される故障が決定される。テスト生成は回路設計後に行われるため、テスト生成後に、状態遷移の設計によって、テスト不能故障と分類された結果を修正することは困難である。

本論文では、回路設計の方法として、コントローラとデータパスを分割して設計する手法に基づくと仮定する。コントローラはどのような演算を行うかを制御信号に出力する。一方、データパスはコントローラの制御信号に基づいて、入力されたデータの処理を行い、データの処理の結果をコントローラへの状態信号としての入力とする。またコントローラは有限状態機械でモデル化されているものとする。このようにコントローラとデータパスは特性が大きく異なるため、それぞれの特性に適した遷移故障検出率を高めるテスト容易化設計手法が求められる。

コントローラの遷移故障検出率を高める評価指標としてQDT値[5]が提案されている。QDT値はFFのQ端子の遷移故障検出条件に基づいたコントローラの状態割当ての評価指標である。QDT値はQ端子の遷移故障検出条件をD端子で満たすと加算され、

値が大きいほど遷移故障検出率が高いと推測できる。

しかしながら、文献[5]ではQDT値が最大となる状態割当てを行えたとしても、遷移故障検出率を最大にできるとは限らない回路が存在していることが確認されている。

そのため、本論文は、コントローラの遷移故障検出率の向上を目的としたコントローラ拡大法を検討する。本論文の構成は、2章で前提知識を述べ、3章で遷移故障検出率向上のためのコントローラ拡大法を説明する。4章で提案手法の結果を示し、最後に5章でまとめと今後の課題について述べる。

### 2. 前提知識

本章では対象故障、テスト方式、対象回路、QDT値を説明する。2.1節で対象故障の遷移故障モデルについて説明する。2.2節で遷移故障モデルに対するテスト生成方式のブロードサイドモデルについて説明する。次に2.3節と2.4節で対象回路のコントローラとその状態割当てを説明する。最後に2.5節でQDT値とその計算方法について例を用いて説明する。

#### 2.1. 遷移故障モデル

遷移故障モデルは、回路中の1つの信号線に仮定され、発生した故障の影響により0から1への遷移もしくは、1から0への遷移時間が増加する故障である。また遷移故障モデルは、故障により増加した遷移時間により故障箇所からFFに伝搬する経路の長さに依存せず観測可能な故障モデルである。0から1に遷移する時間が遅延する立上り遷移故障(slow-to-rise fault : R故障)と1から0に遷移する時間が遅れる立下り遷移故障(slow-to-fall fault : F故障)の2種類の故障が存在する。

遷移故障モデルの検出は2時刻で行う。ある信号線 $a$ の遷移故障の検出条件は、故障の励起条件と故障の伝搬条件の二つを満たすことである。立上り遷移故障について考える。励起条件は1時刻目に信号線 $a$ の値を0に設定し、2時刻目に信号線 $a$ の値を1に設定することである。この設定動作を1時刻目では「初期化」といい、2時刻目は「故障励起」という。この励起条件により2時刻目の信号線 $a$ の値は正常時は1、故障時は0となる。伝搬条件は2時刻目の信号線 $a$ の正常値と故障値のそれぞれの場合において2時刻目の回路応答によるFFの値のいずれかが異なることである。立下り遷移故障も同様に考えることができ、励起条件の時刻における0と1の設定を入れ替えたものである。

#### 2.2. 遷移故障モデルに対するテスト生成方式

遷移故障を検出するためには、遷移前の信号線の値を設定し、遷移後の信号線の値を確かめる必要がある。つまり遷移前、遷移後の2つのテストパターンを連続して印加しなければならない。

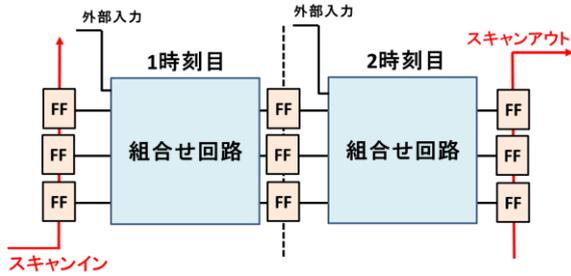


図 1. 本論文のテスト方式を適用した回路モデル

遷移故障モデルに対するテストであるブロードサイド方式では、1 パターン目をシフトイン動作により設定し、1 パターン目の回路応答で 2 パターン目を設定し、2 パターン目に対する応答をキャプチャする。キャプチャした応答をシフトアウト動作によって、外部出力で観測する。この 2 回のキャプチャを実動作速度で行うことで、遷移故障を検出できる。

また、本論文ではブロードサイド方式に加え外部入力に拡張スキャン FF[6]を挿入する。これにより 1 時刻目と 2 時刻の外部入力の値を変更することができる。

各時刻の回路動作を時間展開した回路モデルは図 1 に表される。図 1 の通り 2 パターン目として使われる内部状態は通常動作によって決定される。2 パターン目が通常動作によって設定される利点は、LSI 設計時の制約が少なく、正常な回路を不良と判定する過剰テストによる不要な歩留り低下の可能性が低いことである。

### 2.3. 対象回路

本論文の対象とする RTL 回路はコントローラとデータパスから構成されており、故障検出の対象とするゲートレベルの回路は論理合成によって生成され、フルスキャン設計が適用されている。RTL 回路でコントローラは有限状態機械として設計されている。有限状態機械は入力集合、出力集合、状態集合、出力関数、状態遷移関数で定義されている。この有限状態機械を実現するゲートレベルの順序回路は図 2 のようになる。順序回路は外部入力、外部出力、状態を表現する FF、状態遷移関数や出力関数の論理を実現した組合せ回路で構成される。

### 2.4. 状態割当て

RTL において状態は  $S_0, S_1$  などの記号で区別されている。ゲートレベルにおいて状態は状態変数として 2 進数のビット列で表現される。状態変数のビット数は FF 数である。各状態に数値を対応させる動作を状態割当てと呼ぶ。状態割当てに基づいて、状態変数を

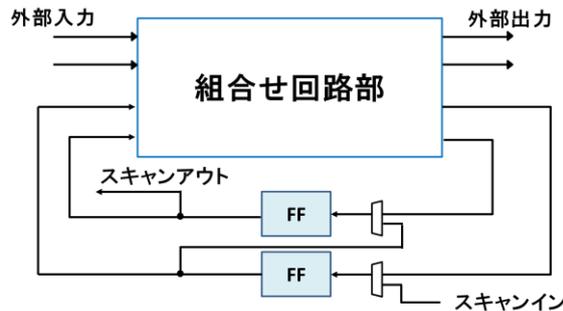


図 2. ゲートレベルの順序回路モデル

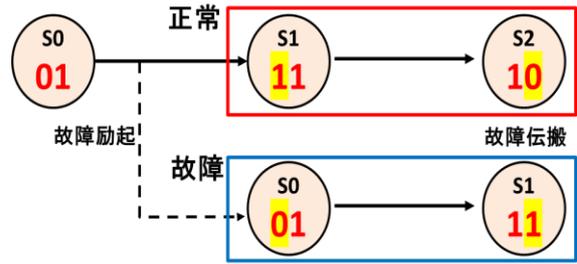


図 3. QDT 値の加算条件の例

FF に割当てて。そのため FF のビット列がコントローラの状態を表す。コントローラは FF 数  $n$  によって  $n$  個の状態変数で最大  $2^n$  個の内部状態を表現でき、 $n!$  通りの割当てがある。

### 2.5. QDT 値

QDT 値はコントローラ回路の遷移故障検出率を RTL の状態割当てで評価する値である。QDT 値は Q 端子の遷移故障を対象としており、テスト方式は 2.2 節のブロードサイド方式を想定している。Q 端子とは順序回路における FF の出力であり、組合せ回路の入力である。反対に D 端子は順序回路における FF の入力、組合せ回路の出力である。

コントローラ回路は順序回路で、状態割当てと状態遷移によって FF の値の遷移が決定する。そのため Q 端子の遷移故障の初期化と故障励起は、コントローラの状態割当てと状態遷移で決定する。つまり Q 端子の遷移故障がテスト可能かどうかは状態割当てで決定する。Q 端子は組合せ回路部の入力であるため、その遷移故障のテストが可能であれば活性化部分は大きいと考えられる。

また、故障影響の観測点が多くなると、活性化部分が大きくなる。観測点は FF で、その入力である D 端子も観測点であるといえる。QDT 値は Q 端子の遷移故障の影響が多くの D 端子に伝搬する状態割当てを求めために用いる。

QDT 値の計算方法を説明する。1 つの FF の Q 端子で生じた故障が 1 つの FF の D 端子に伝搬すれば QDT 値に 1 を加算する。FF 数を  $n$  とすると  $n^2$  通り Q 端子-D 端子対の組合せがある。故障の種類が立上りと立下りの 2 であるため、QDT の最大値は  $2 \times n^2$  である。図 3 に Q 端子の立上り遷移故障の場合の QDT 値加算条件を示す。

QDT 値の計算について例をあげて説明する。FF 数は 2 ビット (FF2, FF1)、状態数は 3 で  $S_0, S_1, S_2$  の順に遷移する。状態はそれぞれ  $S_0(0,1), S_1(1,1), S_2(1,0)$  である。

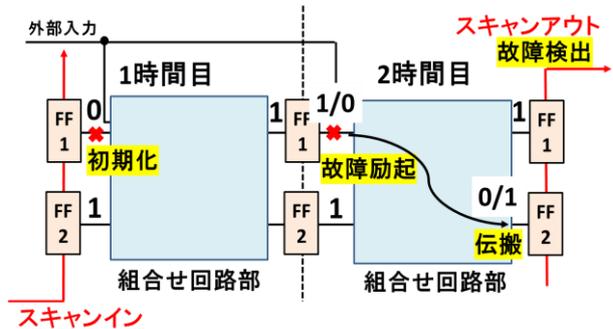


図 4. QDT 値加算例

例では FF2 の Q 端子の立上り遷移故障を考える。まずシフトイン動作で FF に(0,1)を割当てて。2 時間目は  $S_0 \rightarrow S_1$  の通常動作により(1,1)が割当てられる。しかしながら、FF2 の Q 端子立上り遷移故障により  $0 \rightarrow 1$  の遷移が遅延し状態遷移用組合せ回路には(0,1)が入力される。そのため 2 時間目で FF が示す状態は正常時  $S_1$ 、故障時  $S_0$  となる。2 時間目の回路応答も通常動作で決定するため、正常時の動作は  $S_1 \rightarrow S_2$  となり故障時の動作は  $S_0 \rightarrow S_1$  となる。その結果スキャンアウト動作で正常時  $S_2(1,0)$  と故障時  $S_1(1,1)$  となり FF1 に故障が伝搬する。このとき FF2 の Q 端子の立上り遷移故障が FF1 の D 端子に伝搬したとして QDT 値に 1 が加算される。この例を図 4 に示す。

### 3. 遷移故障検出率向上のためのコントローラ拡大法

本章ではサイド FF を QDT 値の評価に加えた、拡張 QDT 値が最大になるようにコントローラの状態遷移を追加する拡大法の説明を行う。

サイド FF は遷移故障のない Q 端子の FF であり、図 4 における FF2 がサイド FF である。このサイド FF の値を 1 時刻、2 時刻で網羅することにより、それぞれ故障の正当化経路と故障の伝搬経路が網羅され遷移故障検出率が向上すると考える。各時刻とサイド FF を網羅した状態遷移が必要になるため、拡張 QDT 値の最大が QDT 値と比較して、 $2 \times n^2 \times 2^{2(n-1)}$  と増大する。それに伴い、拡張 QDT 値の最大化に必要な状態遷移数が  $2^{2(n-1)}$  倍となり、コントローラの状態遷移および外部入力の追加が必要となる。

状態遷移の必要数を例を挙げて説明する。2.5 節の例と同様に FF 数 2(FF1,FF2)の場合を考える。拡張 QDT 値において想定する遷移故障は FF1 の立上り遷移故障、FF1 の立下り遷移故障、FF2 の立上り遷移故障、FF2 の立下り遷移故障の 4 種類である。サイド FF を考慮しなければ、これらの 4 種類の故障を検出できる状態遷移が 4 つ必要になる。サイド FF を考慮すると FF1 の立上り遷移故障の場合、FF2 の 1 時刻目が 0 であり、2 時刻目が 0 である状態遷移のように FF2 の各時刻のパターンが(0, 0), (0, 1), (1, 0), (1, 1)と 1 種類の故障に対して 4 つの状態遷移が必要になる。サイド FF を考慮すると 4 種類の故障  $\times$  サイド FF の 4 つの状態遷移で 16 種類必要になる。FF 数が 2 の場合でも 4 倍の状態遷移が必要になるため、外部入力の追加が必要となる。

状態遷移を追加する手順を説明する。最初に 2.5 節の QDT 値を状態遷移追加前で計算し、追加前のコントローラの状態遷移ペアを列挙する。列挙した状態遷移ペアに不足しているサイド FF を考慮した状態遷移を列挙し、それらを実現する状態遷移ペアをテスト用遷移としてコントローラに追加する。追加した状態遷移を行う外部入力を追加する。図 5 にコントローラ拡大法のアルゴリズムのフローを示す。

### 4. 実験結果

本論文では MCNC ベンチマーク回路[7]の mc を対象回路とする。mc は FF 数 2 ビットの状態数 4 で外部入力 3 ビット、外部出力が 5 ビットである。実験では文献[5]の状態割当てを使用し、外部入力追加

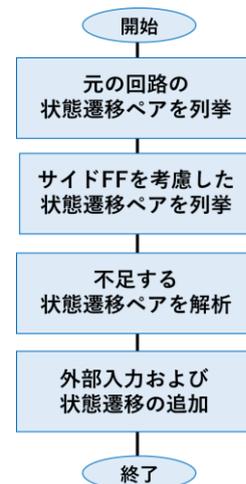


図 5. コントローラ拡大アルゴリズム

表 1. 実験結果

mc	状態遷移 追加前 故障検出率(%)	状態遷移 追加後 故障検出率(%)
mc_0	74.14	83.33
mc_1	75.00	84.21
mc_2	90.00	73.91
mc_3	76.79	77.11
mc_4	90.00	76.27
mc_5	76.79	74.10
mc_6	63.46	82.31
mc_7	72.22	65.49

前の QDT 値、追加後の拡張 QDT 値はともに最大である。FF 数 2 ビットであるため、外部入力追加前の QDT 値の最大が 8、追加後の拡張 QDT 値の最大が 32 である。

外部入力追加前の状態遷移パターン数が 4 で不足している 12 個の状態遷移ペアを追加し、これらの状態遷移を行うよう外部入力を追加した。

mc の各状態割当てに対応した 8 個のフルスキャン設計済み回路を作成し、それぞれ mc\_0 ~ mc\_7 で表す。

論理合成およびフルスキャン設計ツールには、Synopsys 社の Design Compiler と DFT Compiler を使用した。遷移故障のテストパターン生成には、内製の SAT ベースの ATPG を使用した。

表 1 に状態遷移の追加による故障検出率の推移を示す。1 列目は回路名、2 列目は状態遷移追加前の故障検出率であり、3 列目は状態遷移追加後の故障検出率である。

表 1 に示す通り、いくつかの回路で遷移故障検出率を向上できた。しかしながら、状態遷移、外部入力の追加に

による回路面積の増大、複雑化により遷移故障検出率が大きく低下している回路も存在している。

コントローラの状態遷移および外部入力の値は全回路同一であり、状態割当てによる FF の値の遷移は考慮していない。各状態割当てで、遷移故障検出率の変動が生じた理由は、状態遷移および外部入力が同一であることによるものだと考える。

## 5. まとめ

本論文では、コントローラの遷移故障検出率の向上を目的としたコントローラ拡大法として、サイドFFを考慮した状態遷移ペアの作成、外部入力の追加をおこなった。

外部入力追加により遷移故障検出率を多数の回路で向上できた。しかしながら、遷移故障検出率が低下する回路も存在している。

今後の課題は低下した原因を調べ、状態遷移の追加方法や外部入力の割当て方法を改善する。また、状態割当てに応じた状態遷移の追加と外部入力の割当てを実施し、全状態割当て、他のベンチマーク回路の遷移故障検出率の最大化である。

### 参考文献

- [1]. Angela Krstic, and Kwang-Ting (Tim) Cheng, "Delay fault testing for VLSI circuits," Springer Science & Business Media, 2012.
- [2]. Y. Sato, S. Hamada, T. Maeda, A. Takatori, Y. Nozuyama and S. Kajihara, "Invisible Delay Quality SDQM Model Lights Up What Could Not Be Seen," Proceedings of IEEE International Test Conference, Paper 47.1, 2005.
- [3]. 藤原秀雄, "デジタルシステム的设计とテスト," 工学図書株式会社, 2004.
- [4]. J. Savir, and S. Patil, "On Broad-side Delay Test," Proceedings of IEEE VLSI Test Symposium 1994, pp.284-290, Sept., 1994.
- [5]. Masayoshi YOSHIMURA, Yuki TAKEUCHI, Hiroshi YAMAZAKI, Toshinori HOSOKAWA "A State Assignment Method to Improve Transition Fault Coverage for Controllers," The 32nd IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT 2019), Noordwijk, Netherlands, Oct. 2019.
- [6]. S. Dasgupta et al. "An enhancement to LSSD and some applications of LSSD in reliability, availability, and serviceability," International Symposium on Fault-Tolerant Computing, pp. 32-34, 1981.
- [7]. S. Yang, Logic Synthesis and Optimization Benchmarks, Version 3.0, Tech. Report, Microelectronics Center of North Carolina, 1991.