

遷移故障検出率向上のための制御信号の ドントケア割当て法

日大生産工 (学部) ○岩桐 宏成 日大生産工 細川 利典

京産大 吉村 正義

1. はじめに

近年, 超大規模集積回路 (Very Large Scale Integrated Circuits: VLSI) の微細化や高速化, 電源電圧の低下に伴い, 縮退故障モデル[1]のテストだけでなく, 遷移故障モデル[2]などのタイミング欠陥に対するテストが必要不可欠となっている[3]. タイミング欠陥に対する実速度スキャンテストとして, 設計容易性の観点からブロードサイド方式[4]が広く採用されている.

しかしながら, ブロードサイド方式を用いた遷移故障モデルのテスト生成では, 回路構造に起因するテスト不能故障数が縮退故障と比較して数多く存在する. 文献[5]では, 製造を考慮した設計 (Design for manufacturability: DFM) の違反箇所にはテスト不能故障が数多く存在すると述べられている. また, 多重故障は近傍の信号線で発生することが多く, その多重故障に単一故障のテストでテスト不能故障となる故障が含まれている場合, テスト不能故障が検出可能故障の検出を妨げる可能性がある[5]. そのため, 遷移故障モデルのテスト不能故障数を削減し, 故障検出率を向上させることが重要である.

遷移故障モデルのテスト生成における故障検出率向上のテスト容易化設計 (Design for Testability: DFT) [6-8]手法が提案されている. 文献[6, 7]は, テストパターン数削減を目的として, テストポイント挿入法を提案しており, その副次的な効果で遷移故障検出率が向上している. 文献[8]では, レジスタ転送レベル (Register Transfer Level: RTL) において, 無効状態によってテスト活性化用状態を設計してコントローラ拡大を行うことで遷移故障検出率を向上させるようなスキャンテストを対象としたコントローラ拡大を用いたDFT法が提案されている.

文献[8]では, コントローラに対して無効状態において状態または状態遷移の追加設計を行うことで, テスト不能故障数を削減している. 文献[8]の手法による面積オーバーヘッドは最大 8.45%であり, その削減が課題である.

コントローラ拡大のように状態や状態遷移を追加する DFT 手法は大きな面積オーバーヘッドを引き起こす可能性がある. 面積オーバーヘッドを抑制する DFT 手法として, 有効状態遷移時にコントローラからデータパスに供給される制御信号のドントケア割当て手法[9]が提案されている. 文献[9]では, ドントケア割当てアルゴリズムによりできる限り多数の演算器の縮退故障を対象とした並列テストが可能となるように, 制御信号のドントケア割当てを行っている. このアルゴリズムにより, できる限

り多数の状態遷移で演算器のテストを可能とし, 縮退故障のテストパターン数の削減を行っている.

本論文では文献[8]の手法のように無効状態を追加することなく, 機能動作で設計されている有効状態のみを用いたコントローラを使用する. 文献[9]で提案されているドントケア割当て手法を遷移故障のテスト不能故障数の削減に応用する. これにより, 面積オーバーヘッドの増大を抑制しながら, テスト不能故障数を削減し, 遷移故障検出率を向上させることが期待できる. その前段階としてコントローラの各状態遷移のデータパスの動作による解析結果に基づいたドントケア割当てを行い, 故障検出率向上を行う.

2. 諸定義

2.1 遷移故障モデル

遷移故障とは, 信号線における信号伝搬遅延が生じる故障であり, 観測点において信号遷移が所定のクロックサイクル周期内で完了せず, 誤動作が生じる遅延故障モデルの一つである. 遷移故障には, 信号線の 0 から 1 への遷移が遅延する立上り遷移故障 (slow-to-rise) と, 1 から 0 への遷移が遅延する立下り遷移故障 (slow-to-fall) の 2 種類がある.

図 1 に立上り遷移故障の例を示す. 本例の回路は入力 A, B から構成される AND 回路であり, 演算結果がフリップフロップ (Flip-Flop: FF) に出力されるものである. また, A の値は 1 であり, FF で値が観測できるものとする. 1 時刻目において, B の値は 0 であるため FF で 0 が観測される. 2 時刻目において, 正常な回路ではクロックの立上りエッジまでに 0 から 1 へ遷移を完了しており, B の値は 1 である. しかし, 遷移故障が発生した回路では 0 から 1 へ遷移する際に遅延が発生するため, 本来であれば B の値は 1 となるが, 遷移前の値である 0 となる. そのため, 正常な回路であれば FF で 1 が観測されるが, 遷移故障が発生した回路では FF で 0 が観測される.

同様に立下り遷移は, クロックの立上りエッジまでに 1 から 0 への遷移を完了させなければならないが, 遷移故障が発生すると 0 が観測されるタイミングで遷移前の値である 1 が出力され, 観測点での値が正常な回路と異なる可能性がある.

2.2 RTL 回路

本論文で対象とする RTL 回路はデータパスとコントローラに分割して設計された回路であり, フルスキャン設計が適用されている. データパスとコントローラは, データパスからコントローラへの

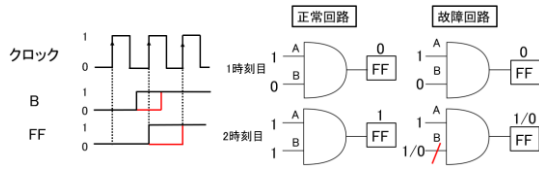


図 1. 立上り遷移故障例

状態信号線と、コントローラからデータバスへの制御信号線で相互に接続されている。図 2 に RTL 回路のデータバスとコントローラの例を示す。

RTL のデータバスにおいて、R0, R1, R2 はホールド機能付きレジスタ、A, B は演算器、M1, M2 はマルチプレクサ(Multiplexer:MUX)、m1, m2 はそれぞれ M1, M2 の制御信号線、r0, r1, r2 はそれぞれ R0, R1, R2 の制御信号線、i は外部入力を示す。本論文では、RTL のコントローラは有限状態機械(Finite State Machine:FSM)でモデル化されていると仮定する。コントローラは、状態集合 {S0, S1, S2} と状態遷移集合 {ST0, ST1, ST2} で構成されている。

また、コントローラから出力される制御信号は、データバスの動作を制御する。例えば、MUX の場合は、複数の入力の中から一つを選択し、ホールド機能付きレジスタの場合は、値を保持するホールドモードと値を取り込むロードモードの二つの動作を制御する。さらに、現状態とデータバスから出力される状態信号によってコントローラの状態遷移が一意に決定される。

2.3 ブロードサイド方式

図 3 にブロードサイド方式[4]のテスト生成モデルを示す。遷移故障を検出するためには 1 時刻目に信号線を初期化し、2 時刻目で初期化した値を基にテストパターンを印加して信号線の遷移を行う必要がある。

ブロードサイド方式は、スキャンベースの遅延テスト手法の一つである。図 4 に 1 時刻目のクロックサイクルでスキャンインによる遷移故障検出の例を示す。a, b, c, d, e, f, g は信号線、a1, a2, b1, b2, b3, c1, c2 は各信号線の分岐先の信号線を示す。また、MUX の制御信号線である m1, m2 において m1 が 0 の場合は b, 1 の場合は a を選択し、m2 が 00 の場合は c, 01 の場合は b, 10 の場合は a を選択する。そして、ホールドモード機能付きレジスタの制御信号線である r0, r1, r2 は 0 の場合ホールドモード、1 の場合ロードモードとなる。ホールド付きレジスタの添字はそれぞれの時刻を表している。図 4 において、信号線 a₁ に立上り遷移故障を定義し、赤色の信号線は信号線 a₁ の立上り遷移故障を検出するために必要な信号線を表している。

信号線 a₁ の立上り遷移故障を検出するためには 1 時刻目において a₁ に初期値として 0 を割当てて必要があるためレジスタ R0₁ に 0 をスキャンインする。a₁ を初期化した後、2 時刻目において 1 時刻目の a, a₁, M1, d, A, f からなる経路を 2 時刻

目の a の値を正当化する経路として用いるために 1 時刻目の M1 で入力として a₁ を選択する必要がある。よって m1 に 1 を割当てる。次に、遷移故障を励起するために 2 時刻目の a₁ に 1 を割当てて必要がある。したがって、1 時刻目の r0 に 1, 2 時刻目の m1 に 1 を割当てる。最後に、遷移故障を検出できるように正当化を行う。本例では、1 時刻目の m2 に 01, R1₁ に 1 を割当てる。以上のように制御信号線に値を割当てることで a₁ の立上り遷移故障を検出することができる。

3. 予備実験

本論文ではデータバス内の故障において、どのハードウェア要素にテスト不能故障が多く存在するかを解析するため予備実験を行った。表 1 に予備実験の結果を示す。対象回路はベンチマーク回路 ex2, ex4, sehwa である。32 ビットのビット幅を持つデータバスの RTL 回路の生成において、動作合成は内製の動作合成ツール PICTHY、論理合成ツールは Synopsys 社の Design Compiler、自動テストパターン生成 (Auto Test Pattern Generator:ATPG) ツールは内製のものを使用した。また、対象故障モデルは遷移故障である。表 1 において、「Circuit」は回路名、「FUNC」は演算器のテスト生成結果、「MUX」は MUX のテスト生成結果、「REG」はレジスタのテスト生成結果を示す。「TF」は対象故障数、「DF」は検出故障数、「UT」はテスト不能故障数、「FC」は故障検出率、「FE」は故障検出効率を示す。

表 1 において、全ての回路の演算器の故障検出率は高いものの、MUX とレジスタの故障検出率は低いという結果が得られた。MUX とレジスタの故障検出率が低下する一因は、以下の二つである。

1) コントローラの設計仕様

2) 論理合成の制御信号のドントケア割当て

一つ目は、コントローラの機能動作に起因するものである。コントローラは状態遷移の仕様に基づいて MUX とレジスタの制御信号を選択するため、設計上、遷移故障をテスト可能にするような MUX、レジスタの入力を選択することが不可能になる場合がある。

二つ目は、論理合成ツールである Design Compiler が実行するドントケア割当てが原因となるものである。論理合成ツールは、回路面積や遅延の最適化を優先する。そのため、論理合成ツールがコントローラに存在するドントケア箇所に自動的に値を割当てるときに遷移故障のテスト可能性を考慮せずに値を割当てている。その結果、特定の連続した MUX、レジスタの入力を選択不可能になる場合がある。一つ目に関してコントローラの機能動作は設計仕様そのものであるため、それに起因するテスト不能故障は回避不可能である。しかしながら、二つ目では、論理合成ツールに起因するドントケア割当てに起因するテスト不能故障は削減可能である。RTL の段階で遷移故障のテスト容易性を考慮してドントケアに値を割当ててことで、テスト

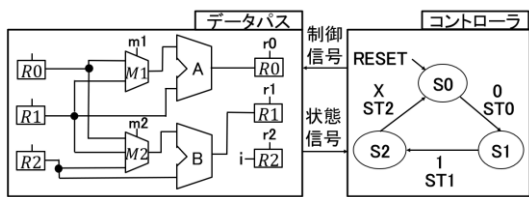


図 2. RTL 回路例

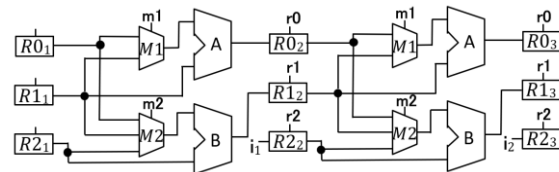


図 3. ブロードサイド方式のテスト生成モデル例

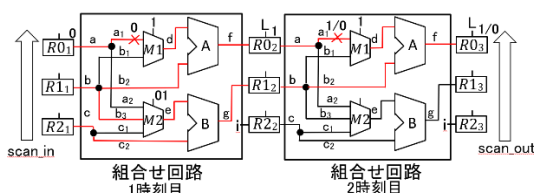


図 4. ブロードサイド方式における立上り遷移故障

不能故障数を削減し、故障検出率を向上させることが可能である。

予備実験結果より、本論文では MUX 及びレジスタのテスト不能故障を削減可能なドントケア割当て手法を検討する。

4. 有効状態におけるドントケア割当て

コントローラの有効状態においては、機能動作に必要なハードウェア要素への制御信号が定義されている。特定の状態遷移において、一部のハードウェア要素の制御信号線の値はドントケアとなる。これらのドントケアは通常、論理合成ツールによって自動的に値が割当てられるが、本来の機能動作に影響を与えることなく、論理合成にドントケアに値を設定することで、追加の制御をデータバスへ制御信号として出力することが可能となる。本論文では、これらのドントケアである制御信号線に対して遷移故障を検出するような論理値を割当てて、このアプローチによってテスト不能故障数を削減し、故障検出率の向上が期待できると考え、ドントケア割当てを行った。

5. 実験結果

本論文では対象回路はフルスキャン設計が適用されたベンチマーク回路 ex2, ex4, sehwa である。32 ビットのビット幅を持つデータバスの RTL 回路の生成において、動作合成は内製の動作合成ツール PICTHY、論理合成ツールは Synopsys 社の Design Compiler、自動テストパターン生成 (Auto Test Pattern Generator: ATPG) ツールは内製のものを使用した。また、対象故障モデルは遷移故障である。

表 1 において、「Circuit」は回路名、「FUNC」は演算器のテスト生成結果、「MUX」は MUX のテスト生成結果、「REG」はレジスタのテスト生成結果を示す。

「TF」は対象故障数、「DF」は検出故障数、「UT」はテスト不能故障数、「FC」は故障検出率、「FE」は故障検出効率を示す。

本論文では、予備実験から MUX の故障検出率が最も低い特徴が確認できたため、MUX のテスト不能故障を減少させることを優先したドントケア割当てを行った。

表 1 と比較すると、ex2 の MUX のテスト不能故障数を 391 個削減し、削減率は、76.37%、レジスタのテスト不能故障数を 1 個削減し、削減率は 0.01%であった。また、ex4 の MUX のテスト不能故障数を 227 個削減し、削減率は 98.69%、レジスタのテスト不能故障数を 32 個削減し、削減率は 100%であった。さらに、sehwa のテスト不能故障数を 215 個削減し、削減率は 56.58%であった。しかしながら、レジスタのテスト不能故障は減少させることができなかった。MUX におけるテスト不能故障の削減率は平均 77.21%であった。故障検出率は、ex2 の MUX は 96.82%、レジスタは 90.94%、ex4 の MUX は 99.88%、レジスタは 100%、sehwa の MUX は 96.71%、レジスタは 91.35%であった。

以上の結果は、コントローラのドントケアに対して適切に値を割当てることにより、MUX のテスト不能故障数を大幅に削減でき、故障検出率を向上させることを示している。また、レジスタに関してもテスト不能故障数を削減できることが確認できる。

表 3 では、Design Compiler によるドントケア割当てから検討手法によるドントケア割当てに手法を変更した際の面積増加率を示す。表から、ex4, sehwa ではそれぞれ 1.698%、1.216%面積が増加したが、ex2 では、1.436%面積を削減していること確認できる。面積の増加が少ないあるいは削減された要因は、Design Compiler によるドントケア割当てにおいて、消費電力を削減するように値が割当てられていたためであると考えられる。

表 1. 予備実験結果

Circuit	FUNC					MUX					REG				
	TF	DF	UT	FC(%)	FE(%)	TF	DF	UT	FC(%)	FE(%)	TF	DF	UT	FC(%)	FE(%)
ex2	35396	35258	138	99.61	100	3410	2898	512	84.99	100	1413	1284	129	90.87	100
ex4	30905	29788	1117	96.39	100	2180	1950	230	89.45	100	1122	1090	32	97.15	100
sehwa	3970	3970	0	100.00	100	4794	4414	380	92.07	100	2960	2704	256	91.35	100

表 2. ドントケア割当て結果

Circuit	FUNC					MUX					REG				
	TF	DF	UT	FC(%)	FE(%)	TF	DF	UT	FC(%)	FE(%)	TF	DF	UT	FC(%)	FE(%)
ex2	26131	26006	125	99.52	100	3811	3690	121	96.82	100	1414	1286	128	90.95	100
ex4	31694	31690	4	99.99	100	2407	2404	3	99.88	100	1154	1154	0	100.00	100
sehwa	3932	3932	0	100.00	100	5009	4844	165	96.71	100	2960	2704	256	91.35	100

表 3. ドントケア割当て手法の変更による面積増加率

	面積増加率
ex2	-1.463
ex4	1.698
sehwa	1.216

6. まとめ

本論文では、遷移故障に対するスキャンテストを対象とし、コントローラの有効状態におけるドントケア割当てを用いることでデータパス部の故障検出率を向上させることが確認できた。

また、本手法では、RTL回路のコントローラのドントケアに対してMUXのテスト容易性向上を主目的としたが、レジスタに関してもテスト不能故障数が削減された。これは、MUXのテスト不能故障数を削減するようにドントケア割当てを行ったことにより、レジスタにおいても遷移故障のテストを行えるパターンが増加したことによりテスト不能故障数が減少したものだと考えられる。

さらに、Design Compilerのドントケア割当ての結果に応じて、面積を削減することができることが確認された。

今後の課題として本手法のドントケア割当てアルゴリズムを提案することが挙げられる。

参考文献

- 1)H. Fujiwara, “Logic Testing and Design for Testability”, The MIT Press, 1985.
- 2)Angela Krstic, and Kwang-Ting (Tim) Cheng, “Delay Fault Testing for VLSI Circuit”, Kluwer Academic Publishers, Boston, 1998.
- 3)Y. Sato, S. Hamada, T. Maeda, A. Takatori, Y. Proc. Nozuyama and S. Kajihara, “Invisible Delay Quality - SDQM Model Lights Up What Could Not Be Seen,” Int. Test Conf., 2005, Paper 47. 1.
- 4)J. Savir and S. Patil, “On Broad-side Delay Test,” Proc. VLSI Test Symp., 1994, pp. 284-290.
- 5)N. Wang, I. Pomeranz, S. M. Reddy, A. Sinha, and S. Venkataraman, “Resynthesis for Avoiding Undetectable Faults Based on Design-for- Manufacturability Guidelines, ” Proc. Design, Automation and Test in Europe Conf., 2019, pp. 1016-1021.
- 6)M. J. Geuzebroek, J. Th. van der Linden, and A. J. van de Goor, “Test Point Insertion for Compact Test Sets,” Test Conference 2000. Proceedings. International, NJ, USA, Oct. 2000, pp. 292-301.
- 7)S. Remersaro, J. Rajski, T. Rinderknecht, Sudhakar M. Reddy, I. Pomeranz, “ATPG Heuristics Dependant Observation Point Insertion for Enhanced Compaction and Data Volume Reduction , ” IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems, MA, USA, Oct. 2008, pp. 385-393.
- 8)Y. Takeuchi, T. Hosokawa, H. Yamazaki, M. Yoshimura, “A Controller Augmentation Method to Improve Transition Fault Coverage for RTL Data-Paths, ” IEEE 25th International

Symposium on On-Line Testing and Robust System Design, Rhodes Island, Greece, July, 2019, pp. 293-298.

9)H. Tokuta, T. Hosokawa, M. Masayoshi, M Arai, “ X-Filling and Test Scheduling Methods for Concurrent Testing Using Optimistically/Pessimistically Structural Symbolic Simulation, ” IEICE Technical Report Online edition: ISSN 2432-6380, July, 2024, Volume 124, Number 148