

コントローラ拡大を用いた遷移故障テストパターン数削減のための演算器のテストレジスタ割当て法

日大生産工(学部) ○竹内 勇希 日大生産工(院) 武田 俊
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、電源電圧の低下にともない、遅延故障のようなタイミング関連の欠陥を検出するためのテストは必要不可欠となっている[1]. また、遅延故障のテスト方式として、設計の容易性の観点から、LoC (Launch-On-Capture) ベース[2] の実速度スキャンテストが広く採用されている。

しかしながら、遅延故障モデルは縮退故障モデルに比べて十分な故障検出を保証するためのテストパターン数が多く、テストコスト増大が問題視されている。したがって、遷移故障モデルに対するテストパターン数削減手法が重要となっている。テストパターン数削減手法としてテスト圧縮法[3-4]やテストパターン数削減のためのテスト容易化設計手法 (Design-for-Testability: DFT)[5-7] が提案されており、多くの故障を並列にテストするテスト並列化によってテストパターン数の削減をおこなっている。

しかしながら、テスト圧縮法において回路構造が原因となり激的にテストパターン数を削減できない可能性がある。また、ゲートレベルにおけるテスト並列化のためのテスト容易化設計手法はゲート数が膨大であるため DFT 適用箇所の探索範囲も非常に膨大である。そのため、テスト容易化設計に膨大な時間を要する可能性がある。また、ゲートレベルで DFT をおこなうと論理合成後の論理の変更により、遅延の増加や論理合成で実行したタイミングの最適性を損失する可能性がある。以上の理由から、ゲートレベル前の抽象度の高いレジスタ転送レベル (Register Transfer Level : RTL) の段階でテスト並列化を考慮してテスト容易化

設計を行うことが重要である。

文献[10]では、標準の設計フローに沿った階層を維持しない演算器のテスト並列化のために、コントローラ拡大とテストポイント挿入を利用した DFT 手法が提案されている。しかしながら、この手法は縮退故障を対象としており、遅延故障を対象としたテストパターン数削減のための方法ではない。

本論文では、スキャンテストにおける遷移遅延故障を対象としたコントローラ拡大を用いた RTL での演算器テスト並列化のための演算器の入出力テストレジスタ割当て法を提案する。

第2章では演算器のテスト並列化のために必要な諸定義を行い、第3章では RTL での演算器のテスト並列化のためのレジスタ割当て問題を解くためのテストスケジューリングの例を示し、第4章では高位合成のベンチマーク回路を用いた実験結果を示し、第5章で結論と今後の課題について述べる。

2. 諸定義

2.1 演算器のテストレジスタ

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から構成されるものとし、実速度スキャンテスト法は LoC を採用する。RTL データパス回路中の演算器 A が他のレジスタを介さずに入力方向もしくは出力方向に到達可能なレジスタを演算器 A のレジスタと定義する。ここで、1 時刻目における演算器 A の左(右)入力から入力方向に到達可能なレジスタを演算器 A の左(右)入力レジスタ、2 時刻目における演算器 A の左(右)入力から入力方向に到達可能なレジスタを演算器 A 左(右)入力の中間レジスタ、2 時刻目の演算器

A Test Register Assignment for Operational Units to Reduce the Number of Transition Fault Test Pattern Using Controller Augmentation

Yuki TAKEUCHI, Shun TAKEDA, Toshinori HOSOKAWA, Hiroshi YAMAZAKI
and Masayoshi YOSHIMURA

A の出力から出力方向に到達可能なレジスタを演算器 A の出力レジスタとする．また，テスト実行時に演算器 A に 1 パターン目のテストパターンを印可するレジスタを演算器 A の入力テストレジスタ，演算器 A に 2 パターン目のテストパターンを印加するレジスタを演算器 A の中間テストレジスタ，演算器 A の 2 パターン目の出力応答を格納するレジスタを演算器 A の出力テストレジスタとする．

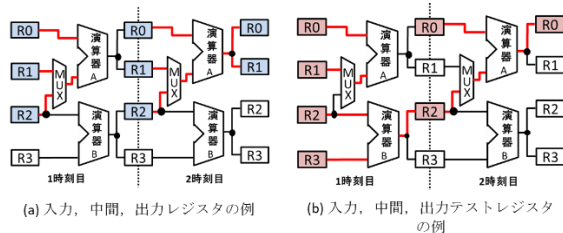


図 2 演算器のレジスタとテストレジスタの例

図 2(a)に RTL データパスにおける入力レジスタ，中間レジスタ及び出力レジスタを示す．1 時刻目の演算器 A に着目すると，1 時刻目において演算器 A の左(右)入力から入力方向に他のレジスタを介さずに到達可能なレジスタは R0, (R1, R2)である．したがって，R0, (R1, R2)は演算器 A の左(右)入力レジスタである．また，2 時刻目の演算器 A に着目すると，1 時刻目と同様に到達可能なレジスタは R0, (R1, R2)である．したがって，R0, (R1, R2)は演算器 A の左(右)中間レジスタである．2 時刻目の演算器 A の出力にも着目すると，R0, R1 が演算器 A の出力レジスタである．

図 2(b)に RTL データパスの部分回路における入力テストレジスタ，中間テストレジスタ及び出力テストレジスタを示す．演算器 A 内の故障についてのテスト生成する際，まず 1 時刻目において，左(右)入力レジスタ R0, (R1)には演算器 A の値を初期化するために 1 パターン目のテストパターンを印加し，出力応答を左中間レジスタ R0 に格納し，左(右)入力レジスタ R2, (R3)には右中間レジスタ R2 を正当化するための値を印加する例を示す．このとき，R0, R1 及び正当化するためのレジスタ R2, R3 は演算器 A の入力テストレジスタとなる．次に 2 時刻目において，左(右)中間レジスタ R0, (R2)から 2 パターン目のテストパターンを印加し，R0 に出力応答を格納する．このとき，左(右)中間レジスタ R0, (R2)は演算器 A の左(右)中間テストレジスタ，出力レジスタ R0 は演算器 A の出力テストレジスタである．

2.2 演算器のテスト集合

本論文では，RTL データパスの演算器 i 単体を対象としてテスト生成を実行したときのテスト集合を，演算器 i のテスト集合と定義する．また，演算器 i のテストパターン数を演算器 i のテスト実行時間と仮定する．本手法ではこの仮定を用いて RTL 回路全体のテスト実行時間を推定する．

2.3 テストレジスタの衝突

RTL 回路中の演算器 A と B の入力テストレジスタが同一のレジスタに割当てられる場合がある．また，中間テストレジスタ割当て，出力テストレジスタ割当てにおいても同様である．このように，複数の演算器のテストレジスタが同一のレジスタに割当てられることをテストレジスタの衝突と定義する．入力レジスタのテストレジスタの衝突を入力テストレジスタの衝突，中間レジスタのテストレジスタの衝突を中間テストレジスタの衝突，出力レジスタの出力テストレジスタの衝突を出力テストレジスタの衝突とよぶ．

3. 演算器のテスト並列化によるテストパターン数削減

3.1 演算器のテストスケジューリング

演算器の並列テストのためのテストレジスタ割当てを最適化するために，演算器のテストスケジューリングを行う必要がある．図 3 の回路を例として演算器のテストスケジューリングを行う．

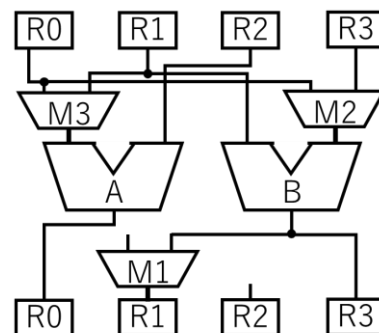


図 3 RTL データパスの例

図 3 の回路は 2 つの演算器 A, B と 4 つのレジスタ R0～R3 で構成される．マルチプレクサ M1 の左入力と R2 の入力は外部入力である．各演算器の入力テストレジスタ，中間テストレジスタ及び出力テストレジスタをテストレジスタの衝突が起こらないように割当てると，表 2 のようなテストレジスタ割当て結果が得られる．

表 2 ex2 の入力テストレジスタ, 中間テストレジスタ及び出力テストレジスタ割当て

演算器名		入力レジスタ 中間レジスタ 出力レジスタ	R0	R1	R2	R3
A	入力	左入力	○	×		
		右入力			○	
	中間	左入力	○	×		
		右入力			●	
B	出力		○			○
	入力	左入力	×			○
		右入力		○		
	中間	左入力	×			○
		右入力		●		
	出力			○		×

表 2 の「○」は入力テストレジスタ, 中間テストレジスタ及び出力テストレジスタとして割当てられたレジスタ, 「●」は 2 パターン目のテストパターンを外部入力から印加するテストレジスタとして割当てられたレジスタ, 「×」はテストレジスタとして割当てられていないレジスタである。例えば, R1 は演算器 B の右入力テストレジスタ, 右中間テストレジスタとして割当てられるが, R1 は外部入力から到達可能なレジスタであるので, 2 パターン目のテストパターンは外部入力から印加するよう割当てて。R2 の右中間テストレジスタも同等に 2 パターン目のテストパターンは外部入力から印加するよう割当てて。表 2 のテストレジスタ割当てを考慮して演算器のテストスケジュールグラフを作成すると図 3 のようになる。縦軸がテストパターン数, 横軸が各レジスタタイプとレジスタ名である。グラフ中の彩色された矩形はテストレジスタに割当てて演算器のテストパターンや期待値を表し, 矩形内の文字は割当てて演算器名とその演算器の入出力の場所を示す。

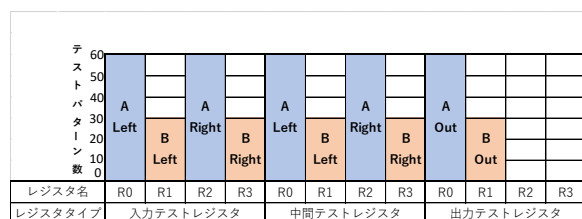


図 3 入力テストレジスタのテストスケジュールグラフ

4.3 コントローラ拡大

通常回路のコントローラは演算器のテスト並列化を実現するように RTL データパスへの制御信号を出力するとは限らない。したがって, 演算器のテスト並列化を実現するためにコントローラ拡大をおこなう必要

がある。コントローラ拡大とは, コントローラをモデル化する状態遷移機械に対して状態や状態遷移を追加する手法である。

コントローラには, 機能動作時には遷移し得ない無効状態が存在する場合がある。本論文ではスキャンテストを前提とするため, コントローラ中のスキャン FF はシフト動作によりテスト時には無効状態にも遷移が可能となる。したがって, 存在する無効状態を用いてコントローラ拡大を行うことで, テスト時にのみ遷移が可能な状態で, 任意の制御信号を出力するように状態遷移を設計することが可能である。このとき, コントローラ拡大をおこなう無効状態を無効テスト状態と呼ぶ。なお, 無効状態が不足する場合はスキャン FF を追加して無効状態を生成する。

本手法では演算器のテストスケジューリング結果を実現する制御信号を出力するように無効テスト状態とその状態遷移を設計する。演算器のテスト並列化を実現する制御信号は, RTL データパス回路中の MUX と, 割当てられたテストレジスタの情報を用いて, 容易に求めることが可能である。

4. 実験結果

本章では, フルスキャン設計が施された ex2, ex4, sehwa, maha の RTL 回路に対して, コントローラ拡大を行わない通常のスキャン設計回路と, 提案手法を適用したフルスキャン設計回路を作成し実験を行い, テストパターン数, 故障検出率, 面積オーバーヘッドを評価した。RTL 回路生成のための動作合成には内製の動作合成ツール PICTHY[11]を使用し, 信号線のビット幅は 16 ビットとした。論理合成ツールは Synopsys 社の Design Compiler を使用し, ATPG も同じく Synopsys 社の TetraMAX を使用し, 対象故障モデルは遷移遅延故障とした。テスト生成のバックトラックス数は 100,000,000 回に設定した。

表 3 に実験結果を示す。「回路名」は実験対象の回路名を示し, 「without」はコントローラ拡大を行わない通常のスキャン設計回路に対する実験結果を示し, 「proposed」は本手法を適用したフルスキャン設計回路の実験結果を示す。「target faults」は対象故障数を示し, 「detect」は検出故障数を示し, 「abort」は打ち切り故障数を示し, 「area」は回路面積を示す。「FC」と「FE」はそれぞれ故障検出率と故障検出効率を示し, 「N_states」はコントローラの有効状態数とコントローラ拡大時に追加した無効テスト状態数の総和, 「TP_num」は回路全体のテストパターン数を示す。

表 3 実験結果

Circuits	without								proposed							
	target faluts	detect	abort	area	FC(%)	FE(%)	N_states	TP_num	target faluts	detect	abort	area	FC(%)	FE(%)	N_states	TP_num
ex2	10627	9945	0	2512	93.58	100.00	6	93	10661	9996	0	2504	93.76	100.00	8	90
ex4	9529	8286	4	2449	86.96	99.95	5	101	9648	8970	0	2484	92.97	100.00	9	103
maha	8217	6477	0	2381	78.82	100.00	26	163	8297	6779	0	2400	81.70	100.00	30	149
sehwa	9185	7171	0	2858	78.07	100.00	26	175	9225	7566	0	2855	82.02	100.00	28	170

ex2, maha, sehwa に対してはテストパターン数を平均 4.89%削減でき, ex4 に対してテストパターン数が 2 個増加するものの故障検出率を 6.01%向上させることができた. 面積は通常のスキャン設計回路に対して平均 0.45%, 最大 1.43%増加した.

5. おわりに

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た演算器のテスト並列化のための RTL での DFT 手法を提案した. 故障検出率が高い回路ではテストパターン数を削減でき, 故障検出率が比較的低い回路では故障検出率を向上させることができた. 今後の課題としてテストパターン数が増加した回路に対する対策, 演算器テストスケジュールの定式化, テストスケジューリングを考慮したテスト生成・圧縮法の提案, 消費電力を考慮したモデルの提案などが挙げられる.

参考文献

- [1] 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," Proc. ITC, p. Paper 47.1, 2005.
- [2] J.Savir and S. Patil, "Scan-based transition test," IEEE Trans. Comput. Aided Design Int. Circuits & Syst., vol. 13, no. 8, pp. 1057-1064, 1994.
- [3] S.Kajihara, I.Pomeranz, K.Kinoshita "Cost-Effective Generation of Minimal Test Sets for Stuck-at Faults in Combinational Logic Circuits," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol.14, Issue12, pp.1496-1504, Dec.1995.
- [4] S Kajihara, I. Pomeranz, K. Kinoshita and S. M.Reddy "On Compaction Test Sets by Addition and Removal of Test Vectors," VLSI Test Symposium, 1994. Proceedings., 12th IEEE, pp.202-207, Cherry Hill, NJ, The USA, Apr 1994.
- [5] 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, pp.292-301, Atlantic City NJ, The USA, Oct 2000.
- [6] Kedarnath J. Balakrishnam and Lei Fang, "RTL Test Point Insertion to Reduce Delay Test Volume," 25th IEEE VLSI Test Symposium (VTS'07), pp.325-332, May.2007.

[7] Jerzy Tyszer, Michal Filipek, Grzegorz Mrugalski, Nilanjan Mukherjee, Janusz Rajski, "New Test Compression Scheme Based on Low Power BIST," 18th IEEE European Test Symposium (ETS 2013), vol.36, pp.683-693, 2017.

[8] Kenichi Yamaguchi, Hiroshi Wada, Toshimitsu Masuzawa, and Hideo Fujiwara, "A BIST Method Based on Concurrent Single-Control Testability of RTL Data Paths," IEEE the 10th Asian Test Symposium (ATS 2001), pp. 313-318, Nov. 2001.

[9] Toshinori Hosokawa, Hiroshi Date, and Michiaki Muraoka, "Two Test Generation Methods Using a Compacted Test Table and a Compacted Test Plan Table for RTL Data Path Circuits," IEICE Trans. on Information and Systems, Vol. E85-D, No. 12, pp.1474-1482, Oct. 2002.

[10] Toshinori Hosokawa, Shun Takeda, Hiroshi Yamazaki, Masayoshi Yoshimura, "Controller Augmentation and Test Point Insertion at RTL for Concurrent Operational Unit Testing," IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS 2017), Thessaloniki, Greece pp17-20, July. 2017.

[11] 石井 英明, 細川 利典, "テスト容易化のためのインターフェースを設けた動作合成システム PICTHY の開発," 第 62 回 FTC 研究会, Jan. 2010.

[12] B.T. Murray and J.H. Hayes, "Hierarchical test generation using pre computed tests for modules," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol.16, no.9, pp.1001-1014, 1990.