

コントローラ拡大を用いたレジスタ転送レベルにおける テストパターン数削減のためのハードウェア要素の テストレジスタ割当て法

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, 超大規模集積回路 (Very Large Scale Integrated Circuits : VLSI) のテストコスト増大に伴い, テストパターン数の削減が重要視されている. テストパターン数削減手法にはテスト圧縮法[1-2]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) [3-6]があり, 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多くのテストパターン数を削減できない可能性が存在する. また, ゲートレベルにおけるテスト並列化のための DFT[3-5]は大規模回路に対して適用する場合, DFT 適用箇所の探索に非常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 を適用する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, 論理合成の適用前の抽象度の高いレジスタ転送レベル (Register Transfer Level : RTL) でテスト並列化を考慮することが重要である.

RTL でのテスト並列化手法のための DFT には階層テスト [7]に基づくもの [8,9]と, 階層を維持しないで設計された回路を対象とするスキャンテストに基づくもの [10]が提案されている. 階層テストは演算器の階層をゲートレベルでも維持することを前提として設計された回路を対象とする. そのため, 回路に対する論理の最適化がなされておらず回路面積や遅延時間が増大するという課題点と, 階層を維持する設計は標準の VLSI の設計フローにそぐわないという課題点がある. また, 階層テスト生成法 [7]で得られたテストプラン [7]と演算器のテストパターンを用いて得られたテスト系列では論理を最適化した回路に対して高い故障検出率を保証できない可能性がある. よって, 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の DFT 手法が重要となる.

テストパターン数削減のための階層を維持しない RTL における DFT 手法として文献 [10]が提案されている. この手法はテストポイント挿入とコントローラ拡大を用いた手法である. しかしながら, この手法には2つの課題点が存在する. 第1に, この手法はテストポイントとしてマルチプレクサ (Multiplexer : MUX) を挿入するため, 回路面積や遅延のオーバーヘッドが増大する可能性がある. 第2に, 演算器以外のハードウェア要素に対する並列テストを考慮していないため, テストパターン数が大きく削減されていない回路が存在する.

上記の課題点を解決するために, 本論文ではテストパターン数削減のためのコントローラ拡大 [11-13]を用いたデータパ

ス中のハードウェア要素である演算器と MUX の並列テストのための DFT 手法を提案する. また, 本手法はフルスキャン設計を適用した回路の縮退故障テストを対象とする. 本手法ではコントローラ中の無効状態に対してハードウェア要素の並列テストが可能な制御信号を明に記述することによってハードウェア要素の並列テストを実現する.

本論文は以下のように構成されている. 第2章ではハードウェア要素のテスト並列化のために必要な諸定義をおこなう. 第3章では RTL でのハードウェア要素のテスト並列化のためのレジスタ割当ての例を示す. 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する. RTL データパスの演算器 j 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 j のレジスタと定義する. ここで, 演算器 j の入力 i から入力方向に到達可能な演算器 j のレジスタを演算器 j の入力 i の入力レジスタ, 演算器 j の出力から出力方向に到達可能な演算器 j のレジスタを演算器 j の出力レジスタとする. また, ある制御信号における演算器 j のテスト実行時に演算器 j の入力 i にテストパターンを印加することが可能なレジスタを演算器 j の入力 i の入力テストレジスタ, その出力応答を観測することが可能なレジスタを演算器 j の出力テストレジスタと定義する. 演算器 j の入力テストレジスタと出力テストレジスタを併せて演算器 j のテストレジスタと定義する.

以下に演算器 j のテスト可能な定義をする. 以下の2つの条件を満たす場合, 演算器 j がテスト可能となる.

- 演算器 j の全ての入力に対して異なるテストレジスタが割当てられている.
- 演算器 j の出力に対して出力テストレジスタが少なくとも1つ以上割当てられている.

2.2 演算器のテスト

本論文では, RTL データパスの演算器 j 単体を対象としてテスト生成を実行したときのテストパターン数を, 演算器 j のテストパターン数と定義する. また, 演算器 j のテストパターン数を演算器 j のテスト実行時間と仮定する. 本手法ではこれによってデータパス全体のテスト実行時間を推定する. 以下に演算器 j のテストの完了を定義する.

A Test Register Assignment Method to Reduce the Number of Test Patterns at Register Transfer Level Using Controller Augmentation

Shun Takeda, Toshinori Hosokawa, Hiroshi Yamazaki, Masayoshi Yoshimura

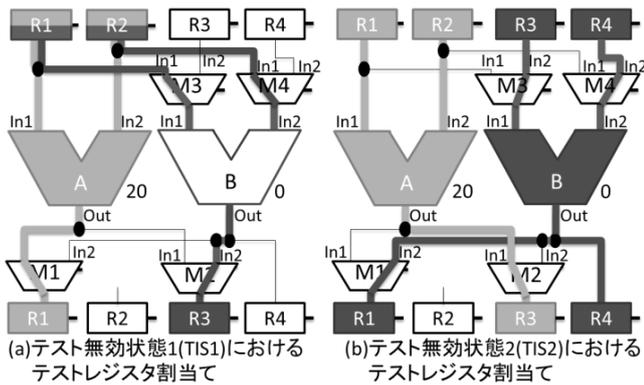


図 3. テストレジスタ割当て

○は MUX の入力テストを示す。

図 3 に、図 2 のテストレジスタ割当ての結果を RTL データパスに反映した結果を示す。演算器の右下の数字が演算器のテストに用いられているテストパターン数を示す。図 3(a) に ITS1 におけるテストレジスタ割当ての結果を示す。演算器 A の入力 In1(In2)の入力テストレジスタは R1, (R2), 出力テストレジスタは R1 である。したがって、演算器 A の入出力テストレジスタが割当てられ、演算器 A がテスト可能となる。また、このテストレジスタ割当てにおいて演算器 A の 20 パターン分のテストと、演算器 A のテストの経路上に存在する MUX M1 の入力 In1 のテストがおこなわれる。演算器 B の入力 In1(In2)の入力テストレジスタは R1, (R2), 出力テストレジスタは R3 に割当てられ演算器 B のテストが可能となる。しかしながら、テストレジスタ割当てにおいて演算器 A と B のテストが可能であるがレジスタ R1, R2 において入力テストレジスタの衝突が発生している。そのため、演算器 B をテストすると全体のテストパターン数が増加してしまうため ITS1 において演算器 B のテストはおこなわれない。しかしながら、演算器 B のテストレジスタを割当てることによって演算器 B のテストが可能となり、テストレジスタと演算器間の経路上の MUX のテストが可能となる。これによって、MUX M3 の入力 In1, MUX M4 の入力 In1, MUX M2 の入力 In2 のテストがおこなわれるため演算器 B のテストレジスタが割当てられている。

図 3(b) に ITS2 におけるテストレジスタ割当ての結果を示す。演算器 A の入力 In1(In2)の入力テストレジスタは R1, (R2), 出力テストレジスタは R3 に割当てられている。したがって、演算器 A の入出力テストレジスタが割当てられ、演算器 A のテスト可能となる。このテストレジスタ割当てにおいて演算器 A の 20 パターン分のテストと MUX M2 の入力 In1 のテストがおこなわれる。また、演算器 B の入力 In1(In2)の入力テストレジスタは R3, (R4)出力テストレジスタは R1, R3 に割当てられている。このテストレジスタ割当てによって、演算器 B の 20 パターン分のテストと演算器 B と MUX M3 の入力 In2, MUX M4 の入力 In2, MUX M1 の入力 In2 のテストがおこなわれる。以上のテストレジスタ割当てによってすべての演算器と MUX のテストがおこなわれる。

4. 実験結果

本章では、フルスキャン設計が施された 12 個の RTL 回路に対して、通常のスキャン設計回路、文献[10]の手法を適用したスキャン設計回路、提案手法を適用したスキャン設計回路を作成し各回路に対して実験をおこない、テストパターン数、面積オーバーヘッドに対して評価した。RTL 回路生成のための動作合成ツールは内製の動作合成ツールを使用し、信号線のビット幅は 32 ビットとした。演算器のテストレジスタ割当て問題を解くために ILP ソルバ SCIP[14]を利用して ITS 集合を求めコントローラ拡大を行った。また、ITS 数が異なる 4 つの入力を ILP ソルバに個別に解かせ、得られた解の中でテストパターン数が最小なものをコントローラ拡大に適用した。4 つの TIS 数は RTL データパス中の入力数が最大の MUX の入力数+0~3 を ITS 数としたものである。SCIP の試行時間は 1000 秒を上限とした。論理合成ツールは Synopsys 社の Design Compiler を使用し、ATPG は同じく Synopsys 社の TetraMAX を使用し、対象故障モデルは単一縮退故障とした。バックトラック制限は 10,000,000 回に設定した。また、故障検出効率が 100.00%に到達しない場合、未検出故障に対してテスト生成のバックトラック制限を 1,000,000,000 回に設定し再度テスト生成をおこなった。

表 1 にテスト生成の結果を示す。「Circuits」は回路名を示し、「without」は通常のスキャン設計回路に対する実験結果を示し、「IOLTS[10]」は文献[10]の手法を用いた実験結果を示し、「Proposed」は提案手法適用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、「ATPGtime」はテスト生成時間(秒)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す。「N_TV」は回路全体のテストパターン数を示す。「*」は規定のバックトラック制限でのテスト生成が終了しなかったためバックトラック制限を 100,000 回に設定しテスト生成をおこなった。「R」は「without」に対するテストパターン数の削減率を示し、「N_ETV」は ILP ソルバから得られたデータパス全体のテスト実行時間の見積もりを示す。

表 2 に回路面積の結果を示す。「Circuits」は実験対象の回路名を示し、「without」は通常のスキャン設計回路に対する実験結果を示し、「IOLTS[10]」は文献[10]の手法を用いた実験結果を示し、「Proposed」は提案手法適用回路の実験結果を示す。「area」は回路面積を示し、「HO」は「without」に対する回路面積の増加率を示し、「N_TPI(MUX)」は制御点と観測点として挿入した 2 入力 MUX 数を示し、「N_addState」はコントローラ拡大時に設計した ITS 数を示し、「N_addFF」はコントローラ拡大時に追加したスキャン FF 数を示す。

テストパターン数はコントローラ拡大を行わない回路と比較して平均で約 33%、最大で約 85%削減することができた。文献[10]の手法を適用した回路と比較して平均で約 14%、最大で約 32%削減することができた。回路規模が大きくなるほど高い効果が得られる傾向が確認できた。また、打ち切り故障が多くテスト生成が困難な FIR_MPEG に対して高い故障検出効率を保証することができた。回路面積オーバーヘッドはコントローラ拡大を行わない回路と比較して平均約 7.11%、最大約 38.37%であった。文献[10]の手法を適用した回路と比較して平均約 0.35%、最大 1.26%であった。

5. おわりに

本論文では、スキャンテストを対象とした回路要素のテスト並列化のためのDFT手法を提案した。

本手法はテストポイントとしてMUXを用いずコントローラ拡大のみで高いテストパターン数削減率を実現した。また、テストレジスタ割当てを求めるためにILPソルバを用いることで短時間での手法適用を実現した。本手法は通常回路に対して平均7.12%の面積オーバーヘッドでテストパターン数を33.47%削減することができた。今後の課題として、提案法に基づくテスト生成・圧縮法の提案、論理合成の最適化の阻害に対する対策、遅延故障モデルへの拡張が挙げられる。

参考文献

- [1] S. Kajihara, I. Pomeranz, K. Kinoshita, "Cost-Effective Generation of Minimal Test Sets for Stuck-at Faults in Combinational Logic Circuits," IEEE TCAD Vol.14 Issue12, pp.1496-1504, Dec.1995.
- [2] 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, NJ, USA, Apr.1994.
- [3] 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, NJ, USA, Oct.2000.
- [4] 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, pp.385-393, MA, USA, Oct.2008.
- [5] M. Yoshimura, T. Hosokawa, and M. Ohta, "A Test Point Insertion Method to Reduce the Number of Test Patterns," IEEE the 11th Asian Test Symposium, pp.298-304, Guam, USA, Nov. 2002.
- [6] Kedarnath J. Balakrishnam and Lei Fang, "RTL Test Point Insertion to Reduce Delay Test Volume," 25th IEEE VLSI Test Symposium, pp.325-332, CA, USA, May.2007.
- [7] B. T. Murray and J. H. Hayes, "Hierarchical Test Generation Using Precomputed Tests for Modules," IEEE TCAD vol.16, pp.1001-1014, Washington DC, USA, Sep.1990.

- [8] S. Ravi, G. Lakshminarayana, and Niraj K. Jha, "High-Level Test Compaction Techniques," IEEE TCAD, Vol.21, pp827-840, July.2002
- [9] T. Hosokawa, H. Date, and M. 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, CA, USA, Oct.2002.
- [10] T. Hosokawa, S. Takeda, H. Yamazaki, M. Yoshimura, "Controller Augmentation and Test Point Insertion at RTL for Concurrent Operational Unit Testing," IEEE IOLTS'17 23rd, pp.17-20, Thessaloniki, Greece, July.2017.
- [11] L. M. Flottes, B. Rouzeyre, L. Volpe, "A controller resynthesis based method for improving datapath testability," IEEE International Symposium on Circuits and Systems, pp.347-350, Geneva, Switzerland, May.2000.
- [12] T. Masuda, J. Nishimaki, T. Hosokawa, and H. Fujiwara, "A Test Generation Method for Datapaths Using Easily Testable Functional Time Expansion Models and Controller Augmentation," IEEE the 24th Asian Test Symposium, pp.37-42, Mumbai, India, Nov.2015.
- [13] S. Ohtake, T. Masuzawa, and H. Fujiwara, "A non-scan approach to DFT for Controllers Achieving 100% Fault Efficiency," Journal of Electronic Testing: Theory and Applications (JETTA) Vol. 16 No. 5, pp.553-566, Taipei, Taiwan, Oct.2000.
- [14] Z. I. Berlin, SCIP (Solving Constraint Integer Programs), <http://scip.zib.de/>, Sep.2017

表 2. 実験結果 2

Circuits	without		IOLTS[10]				Proposed			
	area	OH(%)	N_TPI (MUX)	N_add State	N_add FF	area	OH(%)	N_add State	N_add FF	
ARF	22910	31487	27.24%	0	1	0	31700	27.73%	6	0
BPF	15379	19026	19.17%	0	1	0	19134	19.62%	7	1
ex4	9491	9602	1.16%	1	1	0	9510	0.20%	3	0
ex2	9571	9597	0.27%	0	2	0	9718	1.51%	3	1
dfct	20929	21989	4.82%	1	2	0	21793	3.96%	5	1
DWT_MPEG	33573	33487	-0.26%	0	1	0	33671	0.29%	7	1
FIR_MPEG	36212	36312	0.28%	0	1	0	36392	0.49%	8	1
FFT	23334	23609	1.16%	0	2	0	23669	1.42%	5	1
kim	6556	6595	0.59%	0	2	0	6648	1.38%	5	0
maha	5049	5059	0.20%	1	1	0	5093	0.86%	5	0
sehwa	5878	5879	0.02%	0	1	0	5912	0.58%	4	0
fig17	31074	34526	10.00%	2	2	0	34704	10.46%	8	0

表 1. 実験結果 1

Circuits	without							IOLTS[10]							Proposed										
	target faults	detect	abort	ATPG time(s)	FC(%)	FE(%)	N_TV	target faults	detect	abort	ATPG time(s)	FC(%)	FE(%)	N_TV	R(%)	N_ETV	target faults	detect	abort	ATPG time(s)	FC(%)	FE(%)	N_TV	R(%)	N_ETV
ARF	42501	41477	0	78531.50	97.59	100.00	661	57504	57504	0	5315.58	100.00	100.00	120	81.85	67	57774	57773	1	132217.24	100.00	99.99	97	85.33	68
BPF	28114	28109	4	757420.65	99.98	99.99	446	34041	34041	0	84930.48	100.00	100.00	146	67.26	67	34373	34372	0	7247.25	100.00	100.00	124	72.20	69
ex4	16427	16427	0	965.91	100.00	100.00	85	16685	16685	0	1494.87	100.00	100.00	84	1.18	67	16474	16474	0	790.51	100.00	100.00	79	7.06	80
ex2	17087	17086	1	504274.95	99.99	99.99	87	17100	17098	2	4968.53	99.99	99.99	82	5.75	67	17134	17133	0	4547.49	99.99	100.00	70	19.54	67
dfct	38084	38084	0	800.50	100.00	100.00	112	40158	40158	0	3991.17	100.00	100.00	96	14.29	67	39593	39591	1	97198.89	99.99	99.99	85	24.11	82
DWT_MPEG	57102	57101	0	20542.70	100.00	100.00	163	57291	57291	0	22358.80	100.00	100.00	144	11.66	67	57330	57329	0	4764.54	100.00	100.00	97	40.49	67
FIR_MPEG	61401	58401	2311	104223.45	95.11	96.19	207*	61660	61652	1	546021.00	99.99	99.99	182	12.08	67	61874	61873	0	23580.13	100.00	100.00	177	14.49	67
FFT	42065	42064	1	1961.83	100.00	99.99	182	42405	42402	0	7161.33	99.99	100.00	134	26.37	67	42554	42551	1	62359.40	99.99	99.99	124	31.87	81
kim	10462	10462	0	19.12	100.00	100.00	124	10587	10585	2	697.95	99.98	99.98	101	18.55	36	10656	10652	0	2054.69	99.96	100.00	90	27.42	39
maha	7685	7685	0	0.14	100.00	100.00	188	7711	7710	0	4.26	99.99	100.00	185	1.60	66	7766	7763	0	22.28	99.96	100.00	168	10.64	79
sehwa	8856	8856	0	0.13	100.00	100.00	190	8882	8881	0	851.51	99.99	100.00	181	4.74	66	8924	8924	0	0.88	100.00	100.00	158	16.84	66
fig17	59615	59483	4	78393.19	99.78	99.99	637	65413	65283	84	412762.44	99.80	99.87	430	32.50	161	65878	65856	7	79068.98	99.97	99.99	308	51.65	238