ISSN 2186-5647

2-61

BASTにおけるテストデータ量削減のためのスキャンスライスに基づくインバータブロック構成法

日大生産工(学部)　○錦織 誠　　日大生産工(院)　山崎 紘史　　日大生産工　細川 利典
京産大　吉村 正義　　日大生産工　新井　雅之

１ まえがき

近年の半導体技術の急速な進歩に伴い，大規模集積回路(Large Scale Integrated circuit : LSI)の規模が増大している．一般的にLSIの論理部のテストにはスキャン設計と自動テスト生成ツール(Automatic Test Pattern Generator : ATPG)[1]によるテスト生成が従来からLSI設計者に広く受け入れられてきた．このテスト手法によって高い故障検出率が得られるが，回路の高集積化にともないATPGにより生成されたテストパターン数が増加しており，それに比例してテストデータ量やテスト実行時間が増加する問題が発生している．テストデータ量，テスト実行時間はテストコストに影響する．さらに，テストデータ量はテスタのメモリ容量以下であることが望ましい．そのため，テストデータ圧縮技術の重要性が高まっている．

テストデータ量圧縮の技術として，EDT(Embedded Deterministic Test)[2]，XD-BIST(X-Tolerant Deterministic BIST)[3]，BAST(BIST Aided Scan Test)[4]などが提案されている．本論文では，BASTに着目する．BASTとは，組込み自己テスト(Built-In Self Test:BIST)[4,5,6,7]と，ATPGを組み合わせて高い故障検出効率を維持しながらテストデータ量を削減する技術である．BASTアーキテクチャについては第２章でその詳細を示す．BASTにおいて，疑似ランダムパターン生成器(Pseudo Random Pattern Generator : PRPG)[8,9,10]で生成された疑似ランダムパターン中の特定のビットを，BASTコードと呼ばれるテスタ(Automatic Test Equipment : ATE)のメモリに記憶されたコードに従って反転させる．BASTコードの情報量は，疑似ランダムパターンの反転させるビット数(反転ビット)によって決定されるため，反転ビット数が多いほどBASTコード量が増加する．

疑似ランダムパターンの反転ビット数や反転させないビット数(非反転ビット数)は各決定的パターン，各スキャンチェイン，または各スキャンスライスで異なる．この性質に着目し，スキャンチェインごとの反転ビット数削減手法が提案されている[11]．

本論文では，スキャンスライスごとのBASTコードの反転命令とインバータブロックからの信号により，反転ビット数と非反転ビット数を比較し，少ないビット数を制御する方法を提案する．すなわち，BASTアーキテクチャのインバータブロックの構成を変更することにより，ビット反転するための反転命令数を削減することでBASTコード量の削減を狙う．そのために，本提案手法をISCAS'89, ITC'99ベンチマーク回路に適用し，BASTコード量を評価する．

本論文の構成は以下のとおりである．第２章ではBASTアーキテクチャについて説明し，第３章ではBASTにおけるテストパターン生成法について説明する．第４章では，提案手法であるスキャンスライスごとの反転命令数削減志向のBASTアーキテクチャについて説明し，第５章で実験結果について示したのち，第６章でまとめを述べる．

２ ＢＡＳＴアーキテクチャ

BASTアーキテクチャを図1に示す. BASTは, PRPGとテスト応答圧縮器(Multiple Input Signature Register:MISR)，パターン中の特定のビットを反転させるインバータブロックと，不定状態をマスクするUマスクブロックから構成される．

PRPGで生成されたパターン(PRPGパターン)のみを使用する場合，生成されたパターンは生成されたパターンは疑似ランダムパターンであるため，高い故障検出率が保証されていない．そのためBASTアーキテクチャでは，PRPGパターンを故障検出率が保障されたATPGパターンになるようにビット反転させることで, PRPGパターンでもATPGパターンと同等の故障検出率を保証する．

本論文ではUマスク処理が必要となる回路を扱わないため，以後Uマスクについては説明を省略する．インバータブロックについては2.1節で説明する．

デコーダブロックは，コード化された信号を受け取るためにATEに接続され，インバータブロック中の対応するフリップフロップ(Flip Flop : FF)を制御する．ATEからインターフェースチャンネルに入力されるコード化されたテストパターンを，BASTパターンと呼ぶ．

２．１　インバータブロック

図２に，インバータブロックの回路図を示す．

インバータブロックは，PRPGからスキャンチェインへ入力する信号を，2.2章で示すBASTコードにしたがって反転させる．インバータブロックの出力はXORゲートとFFから構成され, 反転を制御するFFの信号にしたがい，PRPGの出力信号を反転させる．

FFのSetが1，Resetが0のときはPRPGの出力信号を反転し，FFのSetが0，Resetが1のときはPRPGの出力信号がそのままスキャンチェインに印加される．

２．２　ＢＡＳＴコード

An Inverter Block Construction Method Based on Scan Slices to Reduce Test Data Volume on BAST

Makoto NISHIKIORI, Hiroshi YAMAZAKI, Toshinori HOSOKAWA, Masayoshi YOSIMURA and Masayuki ARAI

BAST コードは，線形フィードバックレジスタ(Liner Feedback Sift Register : LFSR)を用いたPRPG で生成される疑似ランダムパターンと ATPG パターンをマッチングしたデータより生成される．

BAST コード例を表 1 に示す．タイムフレームは，スキャンスライスがその時刻に行う処理を示している．反転フラグは，対象となるスキャンスライスでビット反転を行うか否かを示したもので，'1'でビット反転を実行，'0'でビット反転を実行しないものとする．U-マスクフラグは対象となるスキャンスライスで U-マスクを行うか否かを示したもので，'1'で U-マスクを実行，'0'で U-マスクを実行しないものとする．チェインアドレスは，反転・U マスク命令がどのスキャンチェインに適用されるかを 2 進数で表現している．

表 1 におけるスキャンチェインは，チェインアドレスから最大 8 本，スキャンスライスは 3 個存在する．タイムフレーム 1 では，スキャンチェインアドレス '010'(スキャンチェイン 2)に対して U-マスク命令，スキャンチェインアドレス '100'（スキャンチェイン 4)に対して反転命令，スキャンチェインアドレス '001'(スキャンチェイン 1)に対して反転命令を実行する．タイムフレーム 1 の最後で反転命令・U-マスクフラグともに 0 となっているため，シフト命令を実行する．

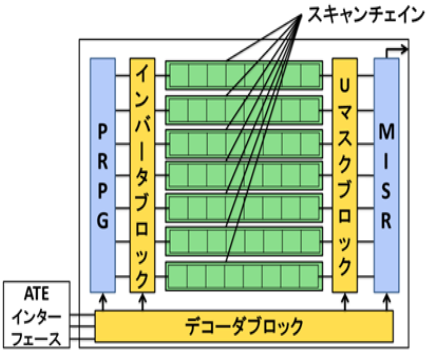

図 1　BAST アーキテクチャ

表 1　ＢＡＳＴコード例

タイムフレーム	1				2	3	
反転フラグ	0	1	1	0	0	1	0
U-マスクフラグ	1	0	0	0	0	1	0
チェインアドレス	0	1	0	0	0	0	0
	1	0	0	0	0	1	0
	0	0	1	0	0	1	0

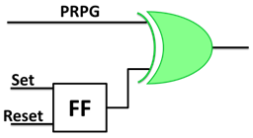

図 2　インバータブロック

3　ＢＡＳＴにおけるテストパターン生成

BASTにおけるテストパターン生成について述べる．ATPGによりテスト対象回路に対するATPGパターン集合TDを生成し，LSI内部に埋め込まれたPRPGが，TDと同一のビット長，テストパターン数のPRPGパターン集合TRを生成する．BASTパターンを，以下のように定義する．

(定義：BASTパターン)

BAST パターンは ATPG パターン $TD = (ppi_1, (ppi_1, ppi_2, \ldots\ldots ppi_n)$ と PRPG パターン $TR = (ppi_1, ppi_2, \ldots\ldots ppi_n)$ をマッチングし，$TB = (ppi_1, ppi_2, \ldots\ldots ppi_n)$を生成する．$ppi_i$は回路中のスキャンFFを表し，$n$は回路中のスキャンFF数を表す($1 \leq i \leq n$)．$TD(ppi_i) \in \{0,1,X\}$は決定的パターン中の$i$番目のスキャンFFの値，$TR(ppi_i) \in \{0,1\}$は疑似ランダムパターン中のi番目のスキャンFFの論理値とする．ただし，Xはドントケアである．

表 2 の BAST パターン生成演算$\cap_s$により，PRPG パターンの値と ATPG パターンの値が一致する場合，または ATPG パターンの値がドントケア(X)の場合，PRPG パターンの値を保持する．また，PRPG パターンの値と ATPG パターンの値が衝突する場合，PRPG パターンの値を反転する．

表 2 に示す$\cap_s$を用いると，TB の i 番目のスキャン FF の値は式(1)で示すことができる．

$$TB(ppi_i) = TD(ppi_i) \cap_S TR(ppi_i) \cdots (1)$$

表２　ＢＡＳＴパターン生成演算$\cap_s$

$\cap_s$		TR 0	TR 1
TD	0	0	0
	1	1	1
	X	0	1

4　スキャンスライスに基づく反転命令とインバータブロック構成

図3に，スキャンスライスに基づく反転命令削減指向インバータブロックを示す．本手法では，従来のインバータブロックにm個のXORゲートと1個のFFを追加する．FF1をビット反転用のFFとし，FF2をスキャンスライス反転用のFFとする．これにより，反転ビット数>非反転ビット数となるスキャンスライスが存在する場合，スキャンスライス中のビットを全て反転させることで，反転ビット数の削減を目指す．

FF1，FF2のXOR演算の結果が 1 の場合はPRPGの出力信号を反転し，XOR演算の結果が0の場合はPRPGの出力信号がそのままスキャンチェインに印加される．

FF2では各スキャンスライスごとに，反転ビット数が非反転ビット数に 1 加えた数より多い場合はSetを1，Resetを0とし，それ以下の場合はSetを0，Resetを1とする．

FF1ではFF2のSetが1，Resetが0の場合にビット反転を行う場合はSetを1，Resetを0とする．ビット反転を行わない場合はSetを0，Resetを1とする．また，FF2のSetが0，Resetが1の場合にビット反転を行う場合はSetを1，Resetを0とし，ビット反転を行わない場合はSetを0，Resetを1とする．

このように設定することで，BASTコードとなる反転命令数が，各スキャンスライスの反転ビット数と非反転ビット数のうち少ないビット数をとる．その結果，

選択したビット数の合計によって反転命令が削減されることで，BASTコード量が削減される．

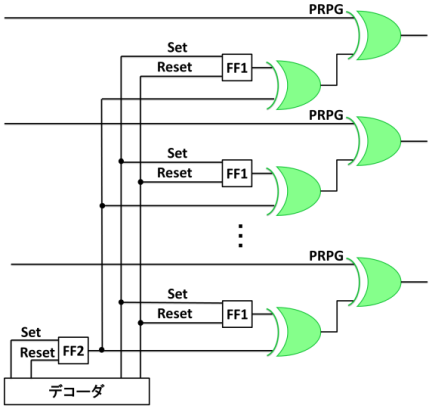

図３　反転命令数削減指向インバータブロック

４．１　提案手法用ＢＡＳＴコード

提案手法用のインバータブロックを実現するにあたり，2.2章で述べたBASTコードを変更する必要がある．

表3に提案手法用BASTコードの例を示す．本論文ではU-マスク処理が必要な回路を扱わないため，U-マスクフラグを反転スキャンスライス指定FFのコントロールフラグ(表3中のFF2制御フラグ)として扱う．本手法ではスキャンスライスを対象としているため，１つのタイムフレーム中でシフト命令が出るまで反転スキャンスライス指定FFのコントロールフラグは変更されない．

表3のタイムフレーム1では，スキャンスライスの値を全て反転し、スキャンチェインアドレス‘011’(スキャンチェイン3)，スキャンチェインアドレス‘101’(スキャンチェイン5)の値をPRPGの値に戻す。タイムフレーム2では、スキャンチェインアドレス‘010’(スキャンチェイン2)，スキャンチェインアドレス‘111’(スキャンチェイン7)，の値を反転させる．その他の命令は，2.2章で述べた命令と同様である．

４．２　スキャンスライス反転の決定

図4にマッチング例を，表4に図4で使用する反転スキャンスライス指定FFの値決定例を示す．

本手法では反転ビット数を重みとして完全二部グラフを作成しているが，提案手法を実現するにあたり，反転ビット数<非反転ビット数の場合は反転ビット数を，反転ビット数>非反転ビット数の場合は非反転ビット数をグラフの重みとして割当てる．

本提案手法では，PRPGパターンとATPGパターンでマッチングを行い，反転ビット数と非反転ビット数を求める．このとき，図4ではスキャンスライス0とスキャンスライス3が反転ビット数>非反転ビット数であるため，反転スキャンスライス指定FF(FF2)のSetを1，Resetを0に設定することで，スキャンスライス0とスキャンスライス3において，PRPGパターンの値は全ビットが反転してスキャンスライスに印加される．したがって，スキャンスライス0とスキャンスライス3では反転命令を出さずにビット反転を行えるため，反転命令数の削減が可能となる．図4の例の場合，提案手法を用いない場合の反転命令数は反転ビット数の合計‘6’となるが，提案手法を用いると反転ビット数は‘2’となる．そのため，従来の反転命令数の1/3でBASTパターンをスキャンチェインに印加できる．

表３　提案手法用ＢＡＳＴコード

タイムフレーム	1			2			3
FF1制御フラグ	1	1	0	1	1	0	0
FF2制御フラグ	1	1	0	0	0	0	0
チェインアドレス	0	1	0	0	1	0	0
	1	0	0	1	1	0	0
	1	1	0	0	1	0	0

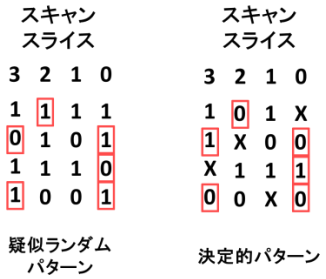

図４　提案手法のビット反転例

表４　反転スキャンスライス指定FFの値決定例

スキャンスライス	反転ビット数	非反転ビット数	反転スキャンスライス指定FFの値	
			Set	Reset
0	3	0	1	0
1	0	3	0	1
2	1	2	0	1
3	2	1	1	0

５　実験結果

ISCAS'89ベンチマーク回路に対して，ドントケア判定[12]とハンガリアンマッチング[13]を用いた従来のBASTアーキテクチャと提案手法での反転命令数削減率と，　BASTコード量削減率の実験・評価を行った．

表5に，スキャンチェイン数を変更した場合の従来手法のBASTと提案手法を用いた場合の反転命令数の比較を示し，図5に提案手法を用いた場合の反転命令数の削減率を示す．表5より，全回路でスキャンチェイン数を変更した場合においても，提案手法を用いた場合は反転命令数を削減できていることがわかる．

表6に，スキャンチェイン数を変更した場合の従来手法のBASTと提案手法を用いた場合のBASTコード量を示し，図6に提案手法を用いた場合のBASTコード量削減率を示す．

表6より，全回路でスキャンチェインを変更した場合においても，提案手法を用いた場合はBASTコード量が削減できていることがわかった．

表５　反転命令数

回路名	テストパターン数	1パターンのビット長	スキャンチェイン本数	BAST[4]	提案手法	削減率
cs13207	274	650	8	7731	6758	12.526
			16	7698	6410	16.732
			32	7719	6364	17.554
cs15850	138	600	8	7416	5708	23.031
			16	6888	5578	19.019
			32	6984	5471	21.664
cs35932	22	1763	8	7790	7024	9.833
			16	7739	7155	7.546
			32	7778	6871	11.661
cs38417	141	1524	8	16869	15375	8.856
			16	16870	15240	9.662
			32	16832	15276	9.244
cs38584	152	1462	8	16380	14460	11.722
			16	16143	14377	10.940
			32	16157	15112	6.468

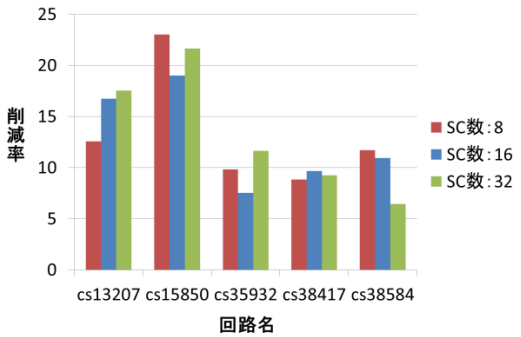

図５　反転命令削減率

表６　ＢＡＳＴコード量

回路名	テストパターン数	1パターンのビット長	スキャンチェイン本数	BAST[4]	提案手法	削減率
cs13207	274	650	8	150995	146130	3.222
			16	113592	105864	6.803
			32	94311	84826	10.057
cs15850	138	600	8	88830	80290	9.614
			16	72792	64932	10.798
			32	67242	56651	15.751
cs35932	22	1763	8	63260	59430	6.054
			16	61086	57582	5.736
			32	63070	56721	10.067
cs38417	141	1524	8	219000	211530	3.411
			16	182436	172656	5.361
			32	165200	154308	6.593
cs38584	152	1462	8	220980	211380	4.344
			16	180762	170166	5.862
			32	162043	154728	4.514

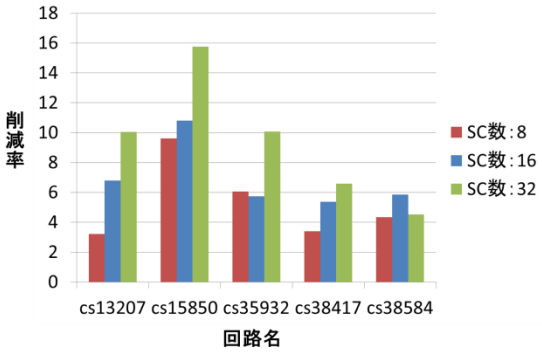

図６　ＢＡＳＴコード量削減率

6　おわりに

本論文では，スキャンスライスの反転命令とその反手命令に対応したインバータブロックを用いたBAST 法を提案し．ISCAS'89 ベンチマーク回路に対して実験を行った．

実験結果から，反転命令数を6.468%~23.031%，BASTコード量を0.193%～8.23%削減することができた．今後の課題として，提案したBAST命令やインバータブロックを制御するデコーダのハードウェアオーバヘッドを評価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] J. Rajski, J. Tyszer, M. Kassab, N. Mukherjee, R. Thompson, K.H. Tsai, A. Hertwig, N. Tamarapalli, G. Mrugalski, G. Eide, and J. Qian, "Embedded Deterministic Test for Low Test Manufacturing Test," in Proc. ITC, pp. 301-310, 2002.

[3] P. Wohl, J.A. Waicukauski, S. Patel, and M.B Amin, "X-tolerant compression and application of Scan-ATPG Patterns in a BIST Architecture," in Proc. ITC, pp. 727-736, 2003.

[4] T. Hiraide, K.O. Boateng, H. Konishi, K. Itaya, M. Emori, H.Yamanaka, and T. Mochiyama, "BIST-aided scan test-A new method for test cost reduction," Proc. VTS, pp. 359-364, 2003..

[5] P. H. Bardell, W. H. McAanney and J. Savir, Built-In Pseudo-Random Testing of Digital Circuits, Chapter 8. John Wiley & Sons, New York, 1987.

[6] Y. Zorian, E. J. Marinissen, S. Dey, "Testing Embedded Core-based System chips," in Proc. IEEE International Test Conference, Washington, DC, pp. 130-143, 1998.

[7] J. Aerts, E. J. Marinissen, "Scan Chain Design for Test Time Reduction in Core-based ICs;" in Proc. IEEE International Test Conference, Washington, DC, pp. 448-457, 1998.

[8] H. Bardell, W. H. McAnney, "Parallel Pseudo-random Sequences for Built-in Test," in Proc. IEEE International Test Conference (ITC), 1984, pp. 302-308, 1984.

[9] H. Bardell, W. H. McAnney, "Pseudo-Random Arrays for Built-In Tests," IEEE Transactions on Computers, Vol. C-35, No. 7, 1986, pp. 653-658.

[10] H. Bardell, W. H. McAnney, J. Savir, Built-In Test for VLSI, Wiley-Interscience, New York, 1987.

[11] M. Tanaka, H. Yamazaki, T. Hosokawa, M. Yoshimura, M. Arai, M. Nakao "An inverter block construction method to reduce test data volume on BAST" IEICE Tech. Rep., vol. 113, no.321, DC2013-51, pp. 171-176, Nov. 2013.

[12]] K. Miyase and Seiji Kajihara, "XID: Don't Care Identification of Test Patterns Circuits and Systems ," IEEE Transactions on Computer-Aided Design

[13] A. Dolan, R. J. Wil son and J. Aldous, Networks and Algorithm :An Introductory Approach, John Wiley & Sons, 1994.