階層 BIST のためのテストライブラリの構築

山口 賢一

Construction of a Test Library for Hierarchical BIST

Ken'ichi Yamaguchi

近年の半導体製造技術の進歩は，VLSI の故障の検出方法にも大きな進歩をもたらしている．筆者らはすでに，高品質なテストが可能であるクロック動作ごとのテスト（Test-Per-Clock Test）と，特別なテスタを必要としない組込み自己テスト（Built-in Self Test）の両方を実現する階層 BIST 法を提案し，ベンチマーク回路を用いた実験により，その有効性を確認している．階層 BIST とは，レジスタ転送レベル回路に対する BIST をレジスタ転送レベルとゲートレベルの２つの階層で行う BIST を総称する概念である．

本稿では，階層 BIST の実用化のための準備段階として、RISC プロセッサや MPEG 回路といった実用的な回路に対して適用するためのテストライブラリの構成法について述べる．具体的には，RISC プロセッサや MPEG 回路に含まれるモジュールである上位ビット出力乗算器および複数の比較器に対するテストライブラリの構築を行った．

1. まえがき

半導体製造技術の急速な進歩によって，VLSI の信頼性を保証するテスト技術が必要不可欠となっている．現在までに様々なテスト方式が提案されているが，現状では故障の有無で出力が異なるような入力系列（テスト系列）を用いるテストが主流である．しかし，論理回路のテスト系列を生成するテスト生成問題は一般に NP 完全であることが知られており，大規模な回路に対して実用的な計算時間ですべての故障に対するテスト系列を求めることはできない．さらに，大規模な回路のテスト系列長は非常に長く，それらを記憶するためには大きな記憶領域が必要になる．そのため，論理回路のテスト費用は増大する傾向にある．

テスト費用の問題を解決するために，テスト生成を行わずにテスト系列を回路内部で発生する方法がある．その方法の一つとして図１で示すテストパターン生成と応答解析を VLSI 上で行う組込み自己テスト（Built-In Self Test: BIST）[1]があり，その重要性が増加している．

一般的な BIST では，テストパターン発生器（TPG）や応答解析器（RA）として，図２に示すような線形フィードバックシフトレジスタ（LFSR）を用いる．LFSR は，複数個のフリップフロップと XOR ゲートから構成でき，フリップフロップの初期値を seed として，また XOR の場所を特性多項式として表現することができ，適切な seed と特性多項式を与えることで，疑似ランダムパターン発生器として利用することが可能である．また、応答を入力することで、XOR を用いた応答圧縮を行い，シグネチャを解析することで応答解析器として利用できる．

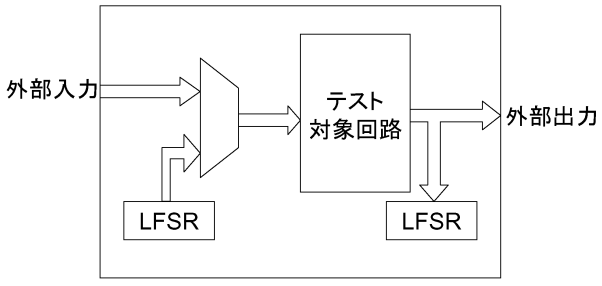


図１　BIST の概念図
Fig.1　Concept of BIST Structure

筆者らは，これまでにレジスタ転送レベル（Register Transfer Level: RTL）に対して階層 BIST に基づく手法を提案している．これらの手法では，回路をテスト容易にするテスト容易化設計（Design for Testability: DFT）

のために加えるハードウェアオーバヘッドが小さく，非常に高い故障検出率を実用時間内で達成することが可能であった．しかし，実験に用いた回路はベンチマーク程度の比較的小規模なものであり，実用化を行うためには，大規模な回路での適用実験が必要である．そこで本稿では，大規模回路としてRISC，MPEGを想定し，階層BISTに基づく手法を適用するために必要となるテストライブラリの構築について述べる．

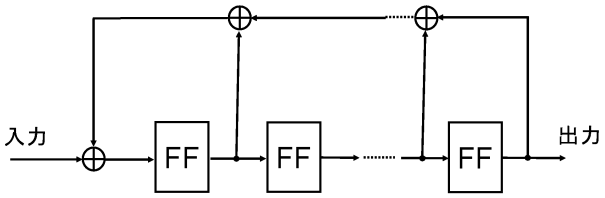


図2　LFSRの概念図
Fig.2　Concept of LFSR

2．レジスタ転送レベル回路

本論文で対象とするRTL回路は，コントローラとデータパスから構成される（図3）．

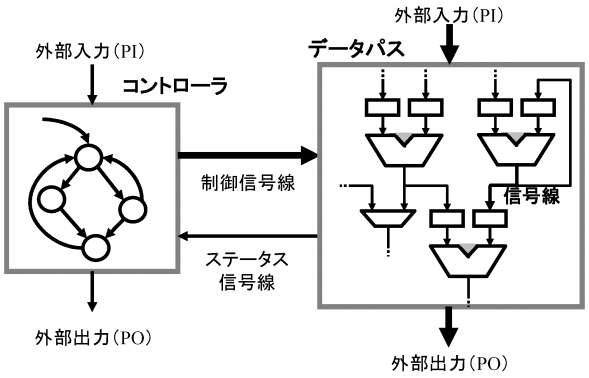


図3　レジスタ転送レベル回路
Fig.3　Register Transfer Level Circuit

コントローラは有限状態機械，データパスは回路要素と回路要素を接続する信号線で記述される．回路要素は，PI，PO，ラッチ，レジスタ，マルチプレクサ，演算モジュール，観測モジュールに分類され，特にマルチプレクサ，演算モジュール，観測モジュールを組合せ回路要素と呼ぶ．各回路要素は端子を持ち，それぞれデータ端子，制御端子，観測端子に分類される．データ端子には，回路要素にデータを入力する入力端子と，回路要素からデータを出力する出力端子がある．制御端子は，コントローラから制御信号を入力する端子であり，観測端子は，コントローラへステータス信号を出力する端子である．

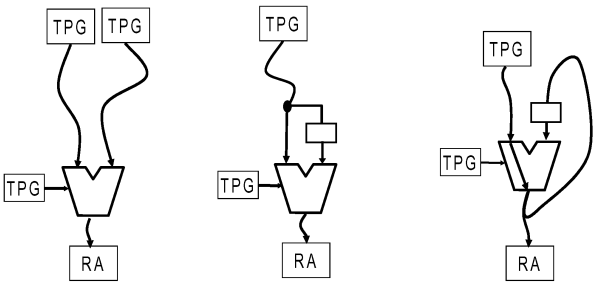
本論文では，様々な組合せ回路要素に対して十分に高い故障検出率を得るためのパターン数およびTPGの性質などに関する考察を実験を通して行い，テストライブラリの構築を行う．

3．階層BIST

階層BISTは，RTLとゲートレベルの2つの階層を利用するBIST方式である．RTLでは，テスト対象となるモジュールに対してTPGからのパターンを印加して，その応答をRAで観測するための経路を生成する．このとき，対象とする故障はゲートレベルでモデル化するため，テスト対象モジュールに対してゲートレベルで故障シミュレーションを行い，故障検出率やテスト実行時間などを評価する．

筆者らは，RTLにおいて時分割単一並行制御可検査性および，それを実現するためのテスト容易化設計法（TCSC法）を提案している[7]．この手法では，TPG，RAをPI，POのみに配置し，ゲートレベルの故障シミュレーションで与えたパターンと同じパターンを与えるために，データパス中の各組合せ回路要素に対して，TPGからのテストパターンの伝搬とRAでの応答の観測のためにテストプランと呼ぶ制御信号時系列を与えることによってテストを行う手法である．ここで，TPGからテスト対象回路要素の入力端子までの経路を制御経路，テスト対象回路要素の出力端子からRAまでの経路を観測経路と呼ぶ．従来の手法(SC法[5]，CSC法[6])では，制御経路と観測経路を図4のtype1のみを用いていたのに対して，[7]では，type1だけではなくtype2，type3の経路も利用するため，ハードウェアオーバヘッドを削減することが可能である．具体的には，従来の手法では，異なるTPGからテスト対象組合せ回路要素の各入力端子へ，および組合せ回路要素の出力端子からRAまでそれぞれ共通部分を持たないようにデータパス上の経路を用いて伝搬する必要があった．そのため，そのような経路が得られない場合は，マルチプレクサなどを挿入する必要があり，結果としてオーバヘッドが大きくなる問題点があった．しかし，TCSC法では，type2のように同じTPGを用いた場合でも，レジスタの段数が異なれば，テスト対象組合せ回路要素へ入力するパターンが異なるためテスト可能であり，type3でも同様にTPGからのパターンがテスト対象モジュールを通過した場合でもテスト可能であることを実験によって示し，制御経路および観測経路として利用した．従って，TCSC法では制御経路と観測経路の生成がSC法[5]やCSC法[6]よりも容易になり，全体のハードウェアオーバヘッドを小さくすることが可能である．

またテストするモジュールの組合せ（テストスケジューリング）の改良によりテスト実行時間も削減することが可能である．SC法では，組合せ回路要素を1つずつテストするためにテスト実行時間が非常に長かった．CSC法では，複数の組合せ回路要素を同時にテストすることを可能にしたが，各組合せ回路要素ごとのテスト実行時間を考慮しなかったため，並行にテストする効果が得られない場合があった．たとえば，テスト実行時間が10クロックのモジュールAとテスト実行時間が1000クロックのモジュールBが同時にテストするようにスケジューリングされた場合，Aのテストが終了しているのにもかかわらず，Bのテストが終了するまでAにパターンが印加され続ける場合などがある．そこで，TCSC法では，各モジュールのテスト実行時間をあらかじめ計算しておき，それを考慮してテストスケジュールを行うことでテスト実行時間の削減を行う．



(a) type1　(b) type2　(c) type3

図4　制御経路と観測経路のタイプ

Fig. 4　Type of control and observation paths

4．テストライブラリ

TCSC法では，各組合せ回路要素に対して目標とする故障検出率を達成するために必要となるテストパターン数をテストライブラリとしてあらかじめ用意する必要がある．文献[7]では，テストライブラリとして，表1の特性を持つ組合せ回路要素に対して，データ入力（DIN）には32bitのLFSRを利用してパターンを与え，制御入力（CIN）には8bitのLFSRの上位bitからパターンを与えた．MUXの制御信号CINは，データ入力の選択のために与える．それ以外のモジュールに関しては，通常動作，右入力端子の値の通過および左入力端子の通過を選択する信号を与える．異なるLFSRの特性多項式は異なるように設定し，各LFSRに対して任意に選んだ5つのseed，特性多項式を用いて故障シミュレーションを行い，検出可能故障に対して故障検出率100％を達成する平均パターン数（# P）とその標準偏差（SD）を求めた（表2）．

表1　組合せ回路特性

Table.1 Characteristics of combinational modules

	DIN	DOUT	CIN	
	#DIN	#DOUT	#CIN	Bit幅
MUX	2	1	1	1
加算器	2	1	1	2
減算器	2	1	1	2
乗算器	2	1	1	2
AND	2	1	1	2
OR	2	1	1	2

表2　テストライブラリ

Table.2　Test Library

	type1		type2		type3	
	#P	SD	#P	SD	#P	SD
MUX	27	3.24	32	2.73	74	1.41
加算器	123	4.85	185	10.15	215	16.64
減算器	167	8.25	298	5.52	325	11.68
乗算器	680	1420	902	12.63	1870	12.86
AND	100	1.22	158	5.10	198	7.53
OR	102	1.58	162	4.70	201	5.48

文献[7]では，表2のテストライブラリを用いてベンチマーク回路Paulin，LWF，Tsengに対してTCSC法を適用し，表3で示す実験結果を得た．

表3では，Wunderlichらの手法[2]，およびCSC法[6]との比較である．Wunderlichらの手法とは，RTL内の閉路に含まれるレジスタを少なくとも1つTPGやRAの機能[3][4]を持たせるように設計変更する手法であり，一度に多くの組合せ回路要素をテストできるので，テスト実行時間が短くなるという利点を持つ．表3でも，すべての回路において，テスト実行時間は一番短くなっている．しかし，レジスタをTPG,RAの機能を有するレジスタへ設計変更するため，ハードウェアオーバヘッドが非常に大きくなる．また，テスト対象回路が大きくなることから，Tsengの場合に見られるように，故障検出率がCSC法やTCSC法に比べて低い．しかし，CSC法やTCSC法では，TPGとRAを外部入出力にのみ配置するので，ハードウェアオーバヘッドは小さい．さらに，同時にテストするパラメタkを与えることで，テスト実行時間と面積オーバヘッドのトレードオフを考慮することができ，TCSC法では，さらに多くのパラメタを導入することで面積優先や時間優先を設計者が選択できる．

表3　TCSC法適用結果
Table.3 Experimental results of TCSC method

		Wunderlichらの手法[2]	CSC法[6]		TCSC法［7］			
					面積優先		時間優先	
			並行度k＝1	並行度k＝2	並行度k＝1	並行度k＝2	並行度k＝1	並行度k＝2
LWF	HW/OH(%)	38.43	21.71	34.66	7.41	8.66	7.41	8.66
	クロック数	106	530	499	616	367	616	367
	FC(%)	100	100	100	100	100	100	100
Paulin	HW/OH(%)	22.49	8.66	14.88	3.14	3.92	4.69	5.47
	クロック数	701	2222	2093	3121	2680	1930	1176
	FC(%)	99.99	99.99	99.99	99.99	99.99	99.99	99.99
Tseng	HW/OH(%)	17.58	17.01	20.00	11.73	12.92	12.01	17.72
	クロック数	657	1843	1568	2386	1567	1758	1025
	FC(%)	99.25	99.99	99.99	99.99	99.99	99.99	99.99

4．テストライブラリの拡張

筆者らは，文献[7]でTCSC法の有効性を確認するために3つのベンチマーク回路LWF，Paulin，Tsengに対して適用実験を行った．本論文では，TCSC法を含む階層BISTをより実用的な回路に対しても適用可能となるようなシステム構成の一環として，テストライブラリの拡張を行う．本論文で新たに対象とする回路はRISCプロセッサおよびMPEG回路である．表4にLWF，Paulin，Tseng，RISCプロセッサおよびMPEG回路のデータパス部の回路特性を示す．＃PI，#POはそれぞれの外部入力および外部出力数を表す．また，|bit|，＃Mod.はそれぞれ，データパスのビット幅，モジュール数を表す．面積は，論理合成ツールAutoLogicII(Mentor Graphics)を用いてゲート数換算で算出した．また，ランダムパターンは，C言語によりLFSRを実装し，周期最大となる特性多項式と任意のシードを与えて生成した．

表4　データパス部の回路特性
Fig.4　Circuit characteristics of datapath

回路	#PI	#PO	|bit|	#Mod.	Area(#gates)
LWF	64	64	32	3	7697.2
Paulin	64	64	32	4	24833.7
Tseng	96	64	32	7	14929.9
RISC	32	96	32	4	58157.9
MPEG	56	148	8	161	69245.5

表4に示す回路において，RISCプロセッサやMPEG回路には表1に示した組合せ回路要素以外にも，上位ビットを出力する乗算器と比較器が含まれている．本稿では，これらのモジュールに対してもテストライブラリの構築を行うために実験を行い，4.1節，4.2節にその結果について示した．これらのモジュールのテストに関しては，リシーディング手法[8]を用いて行った．リシーディング手法とは，図2に示すFF（フリップフロップ）に対して，初期値（シード）とは別に，任意の値を任意のクロックに再度適用する手法である．

4.1　上位ビット出力乗算器に対するテストライブラリの構築

RISCプロセッサには乗算器の上位ビット出力を行うモジュールがある．このモジュールは32ビットのデータ入力を2つ持ち，32ビットのデータ出力を1つ持つ演算モジュールである．このモジュールに対して，ランダムパターンの周期が最大となる特性多項式をもち，任意のシードを与えたLFSRを用いて故障シミュレーションを行った．

しかしながら，50000パターンを印加した時点での故障検出率が85.15％であり，32667個の検出可能な故障のうち，4373個の故障が未検出であった．

そのために，ゲートレベルの回路において高い故障検出率を得るためにLFSRのシードを可変させるリシーディングを行う．リシーディングによって，あらかじめ故障シミュレーションにより分かっている未検出な故障を検出することのできるパターンを与えることで，RISCプロセッサに含まれる乗算器のテストを実用時間内で行う．

実験では，100パターン毎にリシーディングを行うことにより，3400パターン印加時点で99.99%の故障検出

率を達成することができた．この時点で検出できなかった故障はわずか1つで，しかも，故障シミュレーションさえその故障を見つけるパターンを求めることができなかった故障である．

4.2 比較器に対するテストライブラリの生成

RISCプロセッサおよびMPEG回路は，比較器を構成要素として持っている．比較器は，表2に示すモジュールや4.1節の上位ビット出力乗算器とは異なり、観測モジュールに分類される。入力は，32ビットのデータ入力を持ち，観測端子に出力する．
比較器は一般にランダムパターンでテストすることが困難であるとされており，そのための改善方法としてテストポイント挿入手法[9]などが提案されている．しかし，テストポイント挿入法では，論理合成後の回路に対して適用するため，論理合成時に行われた最適化に悪影響を及ぼす．

そこで，本論文では比較器のテストに関してもリシーディングによってテストを行うものとし，テストライブラリの構築を行った．比較器がランダムパターンでテスト困難な原因は，たとえばall1のときのみ1を出力するような比較器では，出力線の0縮退故障を検出するために，入力のすべてが1となるようなパターンが必要となるためである．しかし，リシーディングによってall 1を与えることによって，非常に容易にテストを行うことができる．実験の結果，RISCプロセッサやMPEG回路に含まれる比較器においても，リシーディングにより，わずか数パターンで100％の故障検出率を達成可能であることがわかった．表5に比較器に対する実験結果を示す．ここで，＃REはリシード回数，＃PはFCを達成するのに必要なパターン数，FCは故障検出率を示す．リシードなしの場合に255パターンで終了しているのは，比較器が8ビットモジュールであるため，疑似全数パターンを印加し終えたためである．

表5　比較器に対する実験結果
Fig.5　Experimental Result of Comparators

モジュール	リシードあり			リシードなし	
	＃RE	＃P	FC(%)	＃P	FC(%)
等価比較器	1	100	100	255	0
大小比較器	0	25	100	25	100
定数比較器	1	100	100	255	0

定数比較器はall0のときのみ1を出力する仕様であり，LFSRはシードとしてall0を与えない限り，all0を出力できないために，リシードなしでは故障が検出できなかった．また、等価比較器は，2つのLFSRが同じ周期（255パターン）で動作するため，2つの入力が等価になることが無く，故障が検出できなかった．

実験の結果より，テスト対象回路要素がランダムパターンでテストすることが困難なモジュールであっても，リシーディングを利用して高い故障検出率を得ることができ，テストライブラリを構築できることがわかった．

5．まとめ

本稿では，階層BISTのためのテストライブラリの構築として，より実用的な回路であるRISCプロセッサとMPEG回路を取り上げ，その構成要素である上位ビット出力乗算器と比較器に関して，テストライブラリの構成法を示した．ランダムパターンではテスト困難であるモジュールであるため，リシーディング手法を導入することで実用的な時間で高い故障検出率を達成することができることがわかった．

しかし，現状ではリシーディングによって所望のパターンを印加するためには，図4のtype1の場合においてのみ可能であるため，リシーディングによってデータパス部のオーバヘッドも増加してしまう可能性がある．今後の課題のひとつは，type2，type3の経路を用いた場合のリシーディング手法を考案する必要があると考えられる．また，今回の実験では，上位ビット出力乗算器に関しては，リシードを行う周期を100パターン毎とし，リシーディングによって印加するパターンを未検出故障を検出するパターン中から任意に選択した．リシーディングの増加は，そのシードを保存しておくメモリの増加と，リシードを行う際にシフト動作によって生じる時間の増加が懸念される．今後はリシーディングの周期や順番を考慮することで，リシードによるオーバヘッドと故障検出率やテスト実行時間のトレードオフを考慮した手法の考案がもうひとつの今後の課題である

謝　辞

この研究を行うにあたり，奈良先端大学院大学情報科学研究科コンピュータ設計学講座の施設を借用させていただいた．深く感謝いたします．また，ランダムパターン生成モジュールをコーディングを始め，研究を行うにあたって協力してくれた情報工学科5年の青山瑠美さん，下大園正博くんに感謝します．

文 献

[1]. P. Bardell and W.H. McAnney, “Self-testing of multichip logic modules,” Proc. 1982 IEEE Test Conf., pp.200-203, 1979.

[2]. A.P. Stoele and H.J. Wuderlich, “Hardware-optimal test register insertion,” IEEE Trans. Comput.-Aided Des. Integer. Circuits Syst.,vol.17,no6,pp.531-539,1998

[3]. B. Koenemann, J. Mucha, and G. Zwiehoff, “Built-in logic block observation techniques,” Proc.1979 IEEE Test Conf., pp.37-41,1979.

[4]. L.T. Wang and E.J. McCluskey, “Concurrent built-in logic block observer(CBILBO),” Proc.Int.Symp.on Circuits and Systems,pp.1054-1057,1986.

[5]. 井筒稔，和田弘樹，増澤利光，藤原秀雄，“単一制御可検査性に基づくレジスタ転送レベルデータパスの組込み自己テスト容易化設計法,”信学論（D-I），vol.J84-D-I, no6, pp.527-537, 2002.

[6]. 山口賢一，和田弘樹，増澤利光，藤原秀雄，“レジスタ転送レベルデータパスの単一制御並行可検査性に基づく組込み自己テスト法,”信学論（D-I）,vol.J84-D-I,no6,pp.527-537,2002

[7]. 山口賢一，井上美智子，藤原秀雄，“階層BIST：低いオーバヘッドを実現するTest-per-clock方式BIST,”信学論（D-I），vol.J86-D-I, no7,pp.469-479,2003.

[8]. B. Koenemann, “LFSR-coded test patterns for scan design,” Proc. European Test Conference(ETC), pp.237-242, 1980.

[9]. B. Krishnanmurthy, “A dynamic programming approach to the test point insertion problem,” Proc. ACM/IEEE, pp695-705, 1987.