ISSN 2186-5647

2-41

コントローラ拡大とテストポイントを用いたテスト圧縮効率向上のためのテスト容易化設計

日大生産工(院)　○武田 俊　東工大(院)　大崎 直也　日大生産工　細川 利典
日大生産工　山崎 紘史　京産大・理工　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，テストパターン数削減手法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ポイント挿入[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，テストポイント挿入を用い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(Design-for-Testability:DFT)[3-5]はゲート数が膨大であるためテストポイントの探索範囲も非常に膨大である．そのため，テスト並列化に膨大な時間を要する．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，ゲートレベルに変換される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することが重要である．また, 本手法は VLSI の標準の設計フローに沿った階層を維持しない設計を対象とする．

本論文ではコントローラ拡大とテストポイント挿入を用いたスキャンテストを対象とした RTL での演算器のテスト並列化のための DFT 手法を提案する．

2. 諸定義

2.1 テストポイント

本論文では，制御点のテストポイントとして 2 入力 MUX を用いる．これを疑似外部入力である既存のレジスタに対して接続する．任意の信号線に 2 入力 MUX を経由してレジスタからのパスを追加することで，接続したレジスタからその信号線の値を直接制御可能とする．同様に，観測点としても 2 入力 MUX を使用し，これを疑似外部出力である既存のレジスタに対して接続する．任意の信号線からレジスタへのパスを追加することで，その信号線の値をレジスタで直接観測可能とする．

2.2 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 データパス回路中の演算器 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 の入力から入力方向に到達可能なレジスタを演算器 j 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レジスタを演算器 j の出力レジスタとする．また，RTL 回路のテスト実行時に演算器 j に入力するテストパターンを印可するレジスタを演算器 j の入力テストレジスタ，その出力応答を観測するレジスタを演算器 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．

2.3 演算器のテスト集合

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集合を，演算器 j のテスト集合と定義する．

2.4 テストレジスタの衝突

RTL 回路中の演算器 A と B の入力が同じ入力テストレジスタを割当てられていた場合，一方の演算器のテスト集合で他の演算器のテストを保証できない．そのため，複数の演算器を同時にテスト実行するためには，ぞれぞれの演算器のテスト集合を順番に入力する必要がある．また，出力レジスタも同様である．これ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，出力レジスタの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 テストパターン数見積もり

回路中の全演算器に対してテスト並列化をおこなう場合，全演算器が同時にテストされるために演算器の入出力に対してテストレジスタを 1 対 1 で割当

A Design for Testability Method to Improve Test Compaction Efficiency Using Controller Augmentation and Test Point Insertion

SHUN TAKEDA，NAOYA OHSAKI，TOSHINORI HOSOKAWA，
HIROSHI YAMAZAKI　and　MASAYOSHI YOSHIMURA

てる必要がある．この場合の回路全体のテストパターン数はテストパターン数が最大の演算器のテストパターン数となる．しかしながら，回路構造によっては RTL 回路中のレジスタ数が少ないために，すべての演算器のテスト並列化が実現可能とは限らない．そのため，テストレジスタの衝突が起きるようなテストレジスタ割当てやテストポイント挿入を用いたテストレジスタ割当てをおこなう必要がある．この場合，回路全体のテストパターン数は，レジスタに割当てられた演算器のテストパターン数の最大値となる．

したがって，テストレジスタの衝突を繰返すことによりテストパターン数が増大する．そのため，事前に最小個のテストパターン数を見積もり，テストレジスタ割当て時の制約とする必要がある．

3.3 節に 3.2 節の問題を解決するためのテストポイント挿入について述べる．

3.2 演算器のテスト並列化を実現するための演算器のテストレジスタ割当て

演算器の並列テスト時，演算器の入出力に対してテストレジスタを 1 対 1 で割当てが不可能な場合がある．その場合，見積もりパターン数を超えないようにテストレジスタの衝突を発生させることによって演算器のテスト並列化を実現する．このとき，1 つのレジスタに演算器 j の複数の入力を入力テストレジスタとして割当てないように注意する．

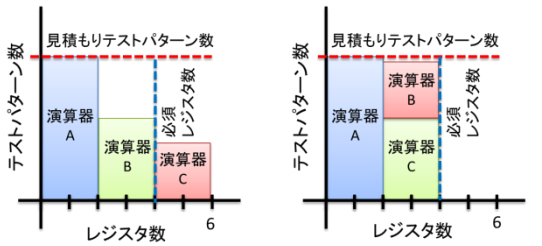

図.2 入力テストレジスタの衝突による レジスタ数の変化

図.2 に入力テストレジスタ衝突による入力テストレジスタ数の変化を示す．図.2 において，演算器の並列テスト実行に必要なテストパターン数の見積もりは演算器 A のテストパターン数と同等であると仮定する．図.2(a)は演算器に割当てられた入力テストレジスタ数がデータパスのレジスタ数を超えるため，演算器のテスト並列化が不可能となる．一方で，図.2(b)は演算器 B と C の入力テストレジスタの衝突を発生された例である．この場合，見積もりパターン数を増大させることなくデータパス中のレジスタ数以内に演算器に割当てるテストレジスタ数を減少させることで．演算器の並列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は演算器とレジスタ間に接続関係が存在している場合のみ可能であるという問題が生じる．

3.3 テストポイント挿入を用いた演算器のテストレジスタ割当て

演算器の並列テストを行う場合，並列テストのためのテストレジスタ割当てをおこなう必要がある．しかしながら，RTL 回路構造上の演算器の入出力とレジスタ間に接続関係が存在しない場合，並列テストのためのテストレジスタ割当てがなされない．そのため，テスト並列化がおこなえずテストパターン数の削減が見込めないことがある．この問題に対して，テストポイント挿入を用いたテストレジスタ割当てをおこなうことで問題を解決する．テストポイント挿入は接続関係のない演算器の入出力とレジスタ間に対してテストポイント(2 入力 MUX)を挿入することで既存の接続関係を維持したまま接続関係を作成する．しかしながら，テストポイントは面積オーバヘッドも大きく，対象故障
数も増加することからテストパターン数が提案手法の削減効果以上に増大する可能性がある．したがって，演算器のテスト並列化を実現するためのテストポイント挿入数は最小化する必要がある．

3.4 問題の定式化

テストレジスタ割当ての定式化をおこなう．また，本手法で扱うデータパスの信号線のビット幅はすべて一様とし，すべての演算器の出力数は 1 とする．

- **テストレジスタ割当て**
  - **定義 1：演算器テスト**

I_j入力 1 出力演算器jのテストパターンを演算器jテストとし，テストパターン数をw_jとする．演算器jのk番目の入力に設定するテスト集合を演算器j入力kテスト集合T_{jk}と定義する．演算器jの出力で観測するテスト応答集合を演算器j出力テスト集合T_jと定義する．

  - **定義 2：演算器テストのスケジューリング**

演算器jの入力kテストと演算器j出力テストは幅 1，高さ 1 の矩形とし，その矩形を w_j 個用いて，それぞれ演算器 j 入力 k テスト集合と演算器 j 出力テスト集合を表現する．演算器テストスケジューリンググラフは，行がテスト実行時刻を示し，列が入力テストレジスタと出力テストレジスタを示すグラフである．演算器jの入力kテストのu番目のテストパターンを入力テストレジスタℓのテスト実行時刻tに配置することを，演算器jの入力kテストuを入力レジスタℓの時刻tにスケジューリングすると定義する．また演算器 j の出力テストの u 番目のテストパターンを入力テストレジスタ l のテスト実行時刻 t に配

置することを，演算器 j の出力テスト u を入力レジスタ l の時刻 t にスケジューリングすると定義する．

➢ **定義 3：諸定義**

演算器$j(j = 1,2,\dots,M)$ ： Mは演算器数

演算器jの入力 $k(k = 1,2,\dots,I_j)$ ： I_jは演算器 j の入力数

レジスタ$\ell(\ell = 1,2,\dots,N)$ ： Nはレジスタ数

テストパターン$u(u = 1,2,\dots,w_j)$ ： w_jは演算器jの入力kテストのテストパターン数

I_{jk} ： 演算器jの入力 k 入力レジスタ集合

O_j ： 演算器jの出力レジスタ集合

T_{jk} ： 演算器j入力 k テスト集合

T_j ： 演算器j出力テスト集合

$S_{jku\ell}$ ： 演算器jの入力kテストuを入力レジスタℓにスケジューリングした時刻

$S_{ju\ell}$ ： 演算器j出力テストuをレジスタℓの出力テストレジスタにスケジューリングした時刻

$X_{jk\ell} \in \{0,1\}$ ： 演算器j入力kテスト集合中の少なくも一つのテストパターンをレジスタℓの入力テストレジスタにスケジュールしたとき，レジスタ$\ell \notin I_{jk}$であれば1，それ以外の場合は0となる．

$Y_{j\ell} \in \{0,1\}$ ： 演算器j出力テスト集合中の少なくも一つのテスト応答をレジスタℓの出力テストレジスタにスケジュー

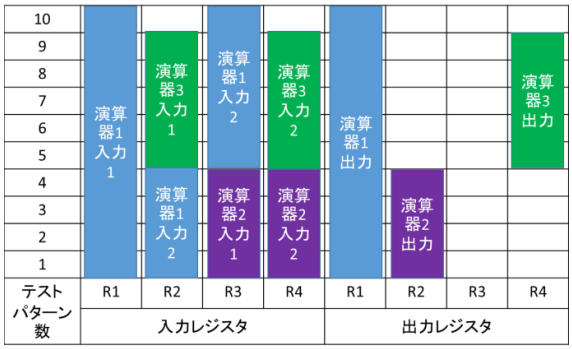

図.3 演算器テストスケジューリンググラフ

ルしたとき，レジスタ$\ell \notin O_j$であれば 1，それ以外の場合は 0 となる．

$g_{jku\ell t} \in \{0,1\}$ ： 演算器jの入力kテストuを入力レジスタℓにスケジューリングした時刻が時刻tである（つまり$S_{jku\ell} = t$）場合 1，それ以外の場合は 0 となる．

$h_{j\ell ut} \in \{0,1\}$ ： 演算器j出力テストuを出力レジスタℓにスケジューリングした時刻が時刻tである（つまり$S_{ju\ell} = t$）場合 1，それ以外の場合は 0 となる．

E_T: 見積もりテストパターン数

[問題定式化]: テストレジスタ割当て

入力 ： データパス中のすべての演算器j入力kテスト集合 T_{jk}の集合 *STI*，演算器j出力テスト集合T_j の集合 *STO*，演算器jの入力kの入力レジスタ集合I_{jk}の集合 *I_REG*，演算器jの出力レジスタ集合O_jの集合 *O_REG*，見積もりテストパターン数E_T

出力 ：*STI* の全ての要素と *STO* の全ての要素がすべてスケジューリングされた演算器テストスケジューリンググラフ

制約 1： $\forall j, \forall k, \forall \ell, \forall u,\ S_{ju\ell} = S_{jku\ell}$

制約 2： $\forall j, \forall k, \forall \ell, \forall u, 1 \le S_{jku\ell} \le E_T$

制約 3： $\forall \ell,\ \forall t,\ 0 \le \sum_j^M \sum_k^{I_j} \sum_u^{w_j} g_{jk\ell ut} \le 1$

制約 4： $\forall \ell,\ \forall t, 0 \le \sum_j^M \sum_u^{w_j} h_{j\ell ut} \le 1$

最適化 ： $\text{Minimum}\left(\sum_{j=1}^{M} \sum_{\ell=1}^{N} \left\{ (Y_{j\ell}) + (\sum_k^{I_j} X_{jk\ell}) \right\} \right)$

図. 3 に演算器テストスケジューリンググラフの例を示す. 左部の数字はテスト実行時刻を示し，「R1」から「R4」はレジスタ名を示し，「入力レジスタ」「出力レジスタ」は入力レジスタと出力レジスタを示す．グラフ中の矩形は各演算器入出力のテスト集合を示す．また，見積もりテストパターン数は 10 である．

3.5 コントローラ拡大

コントローラは演算器のテスト並列化を実現する 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，回路のテスタビリティを向上させるために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，機能動作時には絶対に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 が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に，任意の制御信号を出力するように状態遷移を定義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テスト無効状態と呼ぶ．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．

演算器のテスト並列化を実現する制御信号は，演算器とその入出力レジスタの間に存在する MUX と，割当てられたテストレジスタの情報を用いて，RTL データパスから容易に求めることが可能である．テストポイントが挿入された場合は，テスト無効状態でのみテストポイントが動作するように定義する．また，制御信号の割当てが異なる複数のテスト無効状態が存在することをテスト無効状態の衝突と定義する．テスト無効状態の衝突を削減することでコントローラ拡大に必要な無効状態数とその状態遷移数が削減される．

複数の入力テストレジスタが割当てられた演算器が存在する場合または，出力テストレジスタ衝突が

存在する場合は複数のテスト無効状態を用いる．

4. 実験結果

本章では，フルスキャン設計が施された 13 個の RTL 回路に対して，通常のスキャン設計回路，提案手法適用回路を作成し各回路に対して実験を行い，テストパターン数，面積オーバ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動作合成には内製の動作合成ツール PICTHY を使用し，信号線のビット幅は 32 ビットとした. 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．テスト生成のバックトラック数は 10,000,000 回に設定した ．また，故障検出効率 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数を 1,000,000,000 回に設定し再度テスト生成をおこなった．

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」は通常のスキャン設計回路に対する実験結果を示し，「propose」は本手法適用回路の実験結果を示す.「farget faults」は対象故障数を示し，「detect」は検出故障数を示し，「abort」は打切り故障数を示し，「area」は回路面積を示し，「ATPGtime」はテスト生成時間を示す.「FC」と「FE」はそれぞれ故障検出率と故障検出効率を示し，「ETV」は演算器を並列にテストした際の見積もりテストパターン数を示す.「TP 挿入数」は制御点と観測点の挿入数を示し，「追加状態数」はコントローラ拡大時に追加したテスト無効状態数を示す.「TV 数」は回路全体のテストパターン数を示す。「*」はテスト生成時のバックトラック回数を 100,000 回に設定しテスト生成をおこなった．また，「**」は未検出故障に対する 2 度目のテスト生成テスト生成をおこなっていない．

テストパターン数は通常のスキャン設計回路に対して平均で約 20%，最大で約 82%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．回路面積オーバヘッドは通常のスキャン設計回路に対して平均約 6.5%，最大約 37.4%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．平均 6.5%の面積オーバヘッドでテストパターン数を 20%削減することができた．今後の課題として，回路面積オーバヘッドが 20%以上であった ARF, BPF について，その原因を解析すること，提案法に基づくテスト生成・圧縮法の提案，遷移故障を対象として演算器のテスト並列化のためのテスト容易化設計の提案などが挙げられる．

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 Transactions on Computer-Aided Design of Integrated Circuits and Systems , Vol.14, Issue12, Dec.1995,pp.1496-1504.

[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, Cherry Hill,NJ, The USA, Apr 1994, pp.202-207.

[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,Atlantic City NJ, The USA, Oct 2000 ,pp.292-301.

[4] Santiago Remersaro, Janusz Rajski, Thomas Rinderknecht, Sudhakar M. Reddy, Irith 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, Oct.2008, pp.385-393.

[5] Masayoshi Yoshimura, Toshinori Hosokawa, and Mitsuyasu Ohta, “A Test Point Insertion Method to Reduce the Number of Test Patterns, ” IEEE the 11th Asian Test Symposium (ATS 2002), Nov. 2002, pp.298-304.

表.3 実験結果

Circuits	without								Propose										
	target faults	detect	abort	area	ATPG time(s)	FC(%)	FE(%)	N_TV	target faults	detect	abort	area	ATPG time(s)	FC(%)	FE(%)	N_TPI (MUX)	N_add State	N_TV	N_ETV
ARF	42501	41477	0	22910	78531.50	97.59	100.00	661	57504	57504	0	31487	5315.58	100.00	100.00	0	1	120	67
BPF	28114	28109	4	15379	757420.65	99.99	99.99	446	34041	34041	0	19026	84930.48	100.00	100.00	0	1	146	67
ex4	16427	16427	0	9491	965.91	100.00	100.00	85	16685	16685	0	9602	1494.87	100.00	100.00	1	1	84	67
ex2	17087	17085	3	9571	504274.95	99.99	99.99	87	17100	17098	2	9597	4968.53	100.00	100.00	0	2	82	67
dct	6687	6687	0	4382	0.07	100.00	100.00	33	7128	7128	0	4531	0.04	100.00	100.00	2	2	33	28
dfct	38084	38084	0	20929	800.5	100.00	100.00	112	40158	40158	0	21989	3991.17	100.00	100.00	1	2	96	67
DWT_MPEG	57102	57101	0	33573	20542.7	100.00	100.00	163	57291	57291	0	33487	22358.8	100.00	100.00	0	1	144	67
FIR_MPEG	61401	58401	2311	36212	104223.45	95.11	96.19	207*	61660	61652	1	36312	546021	99.99	99.99	0	1	182	67
FFT	42065	42065	1	23334	1961.83	99.99	99.99	182	42405	42402	0	23609	7161.33	99.99	100.00	0	2	134	67
kim	19462	19462	0	6556	19.12	100.00	100.00	124	10585	10587	2	6595	697.95	99.98	99.98	0	2	101	36
maha	7685	7685	0	5049	0.14	100.00	100.00	188	7711	7710	0	5059	4.26	99.99	100.00	1	0	185	66
sehwa	8856	8856	0	5878	0.15	100.00	100.00	221	8882	8881	0	5879	1002.39	99.99	100.00	0	0	220	66
fig17	59615	59483	4	31074	78393.19	99.99	99.78	637**	65413	65283	84	34526	412762.44	99.87	99.8	2	2	430**	161