7-59

テスト生成高速化のためのアルゴリズム評価

日大生産工（院）○山崎 紘史　　日大生産工　細川 利典
九大　　吉村 正義

1. はじめに

近年，半導体微細化技術の進歩に伴い，大規模集積回路（Large Scale Integrated circuits: LSI）が大規模化・複雑化し，テストパターンの生成時間が増加している．

自動テストパターン生成（Automatic Test Pattern Generation: ATPG）ツールは，論理回路からLSIをテストするテストパターンを自動生成する．LSIのテストでは，品質の高いテストを行うために，高故障検出効率のテストパターンを生成する必要がある．テスト生成アルゴリズムとしては，経路活性化法[1]，充足可能性問題（Satisfiability problem: SAT）[2]，ランダムパターン故障シミュレーションを用いる手法などが過去に提案されている．

経路活性化法を用いたテスト生成法には，テスト生成アルゴリズムの高速化手法として，動的決定順序[3]や衝突による再帰学習[3]などテスト生成中の値衝突の発生を効率的に活用する方法が提案されている．

また充足可能性問題を用いたテスト生成法(Satisfiability problem based ATPG: SAT-ATPG)にも，衝突項の学習[4]や非辞書式バックトラック[5]，ブール強制伝搬(Boolean Constraint Propagation: BCP)[5]の技術により，SAT-solverの処理速度の高速化が図られている．

しかしながら，これらの手法を用いても多大なバックトラックを引き起こす故障が存在し，現実的な時間で100%の故障検出効率を得ることが困難になりつつある．この問題の解決策として，経路活性化法と，充足可能性問題を組み合わせたテスト生成法[6]が提案されている．しかしながら，この手法は経路活性化法でバックトラック数が基準以上発生した故障を充足可能性問題の解法を用いてテスト生成しているので，充足可能性問題の解法の処理でバックトラックを多発して処理時間が大きくなっている場合があると考えられる．

本論文では，経路活性化法，SAT-ATPG，ランダムパターン故障シミュレーションの3つの手法からそれぞれテスト生成容易故障を判定する前段階として，経路活性化法，SAT-ATPG，ランダムパターン故障シミュレーションの全てでテスト生成処理に時間のかかる故障の分類を行った．

2. 経路活性化法を用いたテスト生成

経路活性化法を用いたテスト生成とはPODEMアルゴリズム[1]や，FANアルゴリズム[1]などの回路構造を利用したテスト生成法である．

経路活性化法では含意操作等の値割当てに矛盾が発生した場合，矛盾が生じた箇所に対して異なる値を再度割り当てるバックトラックを行う．このとき判定木において矛盾が生じたノードと，バックトラックを引き起こした原因となるノードが離れた位置関係にあったとしても，バックトラックを引き起こした原因に辿り着くまでバックトラックが行われる．そのためバックトラックを起こした原因をテスト生成中に解析しにくい．また過去に起こした矛盾の原因をテスト生成中に活用しにくいという問題点がある．そのため検出しづらい故障や，一部の冗長故障判定が難しく，これらの故障，冗長故障を判定しようとするとテスト生成時間に多大な影響を与える[7]．

3. 充足可能性問題を用いたテスト生成

SATとは乗法標準形（Conjunctive Normal Form: CNF）が与えられたときに，全ての変数の値を1（真）または0（偽）に定めることで，全体の値を1（真）にできる割当てが存在するか否かを判定する問題である．

CNFを1（真）にできる割当てが存在した場合，充足可能と言い，与えられたCNFが正当化可能であることを示す．CNFが1（真）にできる割当てが存在しない場合，すなわち0（偽）である場合は与えられたCNFが正当化不可能であることを示す．

過去にSATを用いたテスト生成法[2]が提案されており，充足可能の場合はそのときの外部入力の変数値の1（真）または0（偽）の割当てがテストパターンとなる．また充足不可能だった場合は冗長故障と判定される．

SATでは変数の値割当て後に，その割当てられた変数から強制的に値が決まる変数に値を割当てる処理(BCP)を行う．このとき変数値に矛盾が発生した場合バックトラックを行う．最新のSAT-solver[8]ではCNFに含まれる変数に値を割当てる処理で，値割当て頻度の高い状態変数から割当てる変数選択ヒューリスティックVSIDS[4]や，値割当てに矛盾が生じた場合，矛盾を解析・学習[5]する処理が組み込まれている．

SATは経路活性化法のように回路構造から値を決定していくのではなく，CNF内の項(Clause)に含まれる変数の使用頻度に応じて値を決定する．SATはCNFの構造上，項に含まれる変数の1つでも0（偽）になった時点で矛盾と判定できる．そのため矛盾の早期発見や，高速な冗長判定が可能である．また経路活性化法で多大なバックトラックののち打ち切られる故障に関しては，SATは高速なテスト生成が可

Evaluation of Algorithms for Acceleration of Test Generation

Hiroshi YAMAZAKI, Toshinori HOSOKAWA, and Masayoshi YOSHIMURA

能である[7].

しかしながら，全ての故障を SAT でテスト生成した場合，充足可能判定に非常に時間がかかる故障が存在し，テスト生成時間に多大な影響を与える．

4. ランダムパターンを用いたテスト生成

テスト生成において，多くの故障はランダムに生成したテストパターンで検出される確率が高い．しかしながら，高い故障検出率を望む場合にはランダムパターンだけでは効率が悪い．また疑似乱数を用いるため，本来検出が困難であるにもかかわらず，偶発的に発見される故障も存在する．本研究では，多数のランダムパターンで検出可能な故障と，ランダムパターンで偶発的に検出される故障を分類するためにランダムパターン故障シミュレーションを利用する．

5. 経路活性化と SAT を組み合わせたテスト生成

過去に経路活性化法を用いたテスト生成アルゴリズムである FAN アルゴリズムと，SAT-ATPG を組み合わせた手法[6]が提案されている．この手法はバックトラックリミット 100 の FAN アルゴリズムで打ち切られた故障に対して，再度 SAT-ATPG を行う．しかしながら，FAN アルゴリズムで打ち切られた故障の中には，バックトラックリミットの微量の増加でテスト生成が行える故障も含まれる．そのため，FAN アルゴリズムのほうが高速にテスト生成を行える故障も，SAT-ATPG でテスト生成を行っており，テスト生成時間に無駄が生じている可能性が有る．5.1 でこの問題を考慮したテスト生成アルゴリズム(ハイブリット ATPG)の全体アルゴリズムを示す．また 5.2 で，各テスト生成法で多大なテスト生成時間を要する故障を分類する方法を示す．

5.1 ハイブリット ATPG

図 1 にテスト生成アルゴリズムを考慮したハイブリット ATPG の全体アルゴリズムを示す．

(Step1)
テスト生成対象回路データを読み込み，代表故障集合を作成する．
(Step2)
Step1 で作成した代表故障集合に対して，ランダムパターン故障シミュレーションを行う．
(Step3)
Step2 で未検出だった代表故障に対して，経路活性化法 ATPG を行う．このときバックトラックリミットは 0 から 10 程度の少ない値とする．
(Step4)
Step3 の結果，打ち切り故障が存在しなければ終了，それ以外は Step5 へ進む．
(Step5)
Step3 で打ち切られた故障に対して，どの ATPG 法が一番容易にテスト生成を行えるのか解析し，分類をする．
(Step6)
Step5 で分類された故障ごとに再度テスト生成を行う．

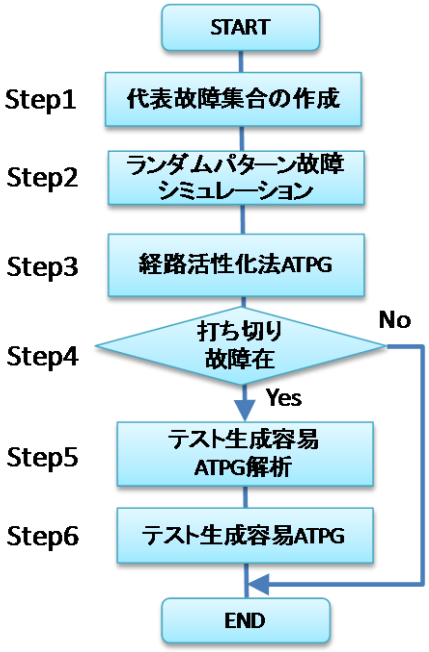

図 1. ハイブリット ATPG 全体アルゴリズム

5.2 テスト生成困難故障分類

5.1 で述べた，ハイブリット ATPG を実現するには，各テスト生成法でテスト生成が容易な故障(容易故障)と，困難な故障(困難故障)を分類し，解析する必要がある．図 2 に容易故障と困難故障を分類する方法を示す．

まず代表故障に対してランダムパターン故障シミュレーションを行う．このとき疑似乱数により偶発的に検出された故障を，困難故障に含まないように検出回数 n を指定する．
生成したランダムパターンで，n 回検出未満の故障をランダムパターン検出困難故障，n 回検出以上の故障をランダムパターン容易故障と定義する．また n の値は， 100 から 1000 程度の値が望ましい．

次にランダムパターン検出困難故障を対象に，経路活性化法でテスト生成を行う．このとき低バックトラックリミット(0 から 10 回程度)でテスト生成を行い，打ち切られた故障を経路活性化困難故障，それ以外を経路活性化容易故障と定義する．

最後に経路活性化困難故障を対象に，SAT-ATPG を行う．このとき SAT-solver の値衝突の回数に制限を設けて(100 から 1000 回程度)，打ち切られた故障を ATPG 困難故障，それ以外を ATPG 容易故障と定義する．SAT-solver は 3 で述べたように矛盾の早期発見が可能である．同じ時間で実行した場合 SAT のほうが経路活性化法より多くのバックトラックを行える．そのため SAT-ATPG の値衝突制限値は経路活性化法よりも大きい値としている．

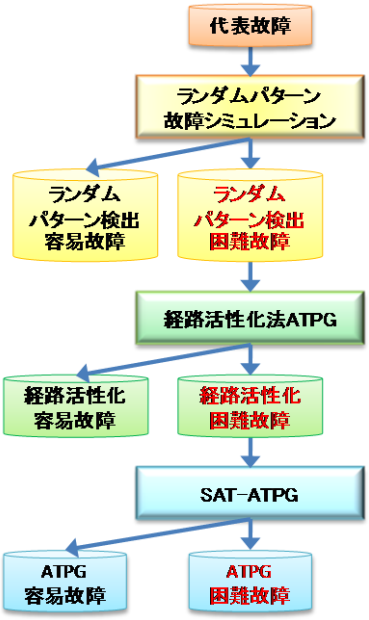

図 2. テスト生成困難故障分類方法

6. 実験結果

本論文では実験結果として 5.2 で述べたテスト生成困難故障の分類を行い，ランダムパターン検出困難故障，経路活性化困難故障，ATPG 困難故障を分類した．実験環境は，CPU は CoreDuo T2300(1.6GHz)，メモリは 1GB，OS は Windows XP，使用した SAT-solver は MiniSat-C v1.14.1[8]である．実験対象として ITC'99 ベンチマーク回路の b14，b15，b20，b21，b22 を組合せ回路化したものを用いた．故障モデルは単一縮退故障である．また経路活性化法，SAT-ATPG では故障シミュレーションは使用していない．

表 1 にランダムパターン検出困難故障の実験結果，表 2 に経路活性化困難故障の実験結果，表 3 に ATPG 困難故障の実験結果を示す．

ランダムパターン検出困難故障の検出回数 n は 100 回，500 回，1000 回の 3 種類とした．経路活性化困難故障は，ランダムパターン検出困難故障の 100 回未満検出の故障を対象に，バックトラックリミット 0 回，5 回，10 回の 3 種類で実験を行った．またテスト生成アルゴリズムは SPOP[9]を利用した．ATPG 困難故障は，経路活性化困難故障のバックトラックリミット 5 回で打ち切られた故障を対象に，SAT-solver の Conflicts リミットを 100 回，500 回，1000 回の 3 種類で実験を行った．また経路活性化困難故障の検出故障，冗長故障の割合を調べるため Conflicts リミット無制限での実験も行った．

表 1 より b15 以外は，検出回数 n が 100 回と 1000 回の故障検出率は 5%程度しか変化しないことがわかった．

表 2 よりランダムパターン検出困難故障のほとんどが，低バックトラックリミットの経路活性化法でテスト生成可能であることがわかる．またテスト生成時間と打ち切り故障数を考慮するなら，バックトラックリミット 10 回が最も効率が良いことがわかる．

表 3 より Conflict 数と打ち切り故障数を考慮するなら，Conflicts リミット 500 回程度が良いことがわかった．

また打ち切り故障数の多い b15 では，Conflict リミット無制限でテスト生成を行った結果，他の回路より冗長故障が多く含まれていることがわかる．SAT-ATPG は経路活性化法より高速な冗長判定が可能と言われているが，SAT-ATPG でも判定の難しい冗長故障があることがわかった．

今回は Conflict 数で実験を行ったが，SAT は Conflicts 数と，実行時間の間には必ずしも比例の関係にあるとは言えない．そのため今後は SAT-solver の実行時間で容易故障，困難故障と判定する方法に変更して，再度実験を行う予定である．

7. おわりに

今回，ランダムパターン故障シミュレーション，経路活性化法，SAT-ATPG の全てで，テスト生成が困難な故障の分類を行った．

今後の課題として，分類した ATPG 困難故障の得意，不得意とする ATPG 法の解析が挙げられる．また FANⅢ[10]や FIRE[11]などのテスト生成法も組み合わせた困難故障集合の作成や，ITC'99 ベンチマーク回路の b18,b19 などの大規模回路による困難故障集合の作成も挙げられる．

参考文献

1) 藤原秀雄，“ディジタルシステムの設計とテスト”，工学図書株式会社（2004）,pp1-262

2) Paul Tafertshofer，Andreas Ganz，”SAT Based ATPG Using Fast Justification and Propagation in the Implication Graph”，IEEE/ACM international conference on Computer-aided design(1999), pp139-146

3) Chen Wang，Sundhakar M.Reddy “Conflict Driven Techniques for Improving Deterministic Test Pattern Generation”，IEEE/ACM international conference on Computer-aided design (2002),pp87-93

4) Joao P. Marques-Silva，“GRASP: A Search Algorithm for Propositional Satisfiability”，IEEE TRANSACTIONS ON COMPUTERS Volume 48 (1999), pp506-521

5) Matthew W. Moskewicz，“Chaff: Engineering an Efficient SAT Solver”，38th annual Design Automation Conference (2001), pp530-535

6) Drechsler，R. Eggersgluss，S. Fey，G. Glowatz，A. Hapke，F. Schloeffel，J. Tille，D，“On Acceleration of SAT-Based ATPG for Industrial Designs” Computer-Aided Design of Integrated Circuits and Systems Vol27 (2008), pp1329-1333

7) 秋山祐介,”充足可能性問題の解法を用いた ATPG 困難故障の故障分類判定の高速化に関する研究”，日本大学大学院生産工学研究科 2008 年度修士研究(2009)

8) Niklas Een, Niklas Sorensson “An Extensible SAT-solver” , Lecture Notes in Computer Science Vol2919(2004) , pp333-336

9) M.Henftling, H.Wittmann and K.Antreich , A Single-Path-Oriented- Fault-Effect Propagation in Digital Circuits Considering Multiple-Path Sensitization , Proceedings of ICCAD(1995), pp304-309.

10) H.Fujiwara , T.Shimono, “On the Acceleration of Test Generation Algorithms”, IEEE Transactions on Computers Vol32 (1983), pp1137-1144

11) Iyer, M.A. Abramovici, M.Dept, “FIRE: a fault-independent combinational redundancy identification algorithm” , Very Large Scale Integration (VLSI) Systems Vol4 (1996), p295-301

表1. 実験結果 ランダムパターン検出困難故障

回路名	ランダムパターン数	代表故障数	未検出故障数			故障検出率(%)		
			検出回数100回未満	検出回数500回未満	検出回数1000回未満	検出回数100回以上	検出回数500回以上	検出回数1000回以上
b14	10000	22802	19371	20404	20637	15.05	10.52	9.49
b15	10000	21988	15407	18004	18986	29.93	18.12	13.65
b20	10000	45459	38407	40576	41035	15.51	10.74	9.73
b21	10000	46154	38970	41239	41714	15.57	10.65	9.62
b22	10000	67536	57074	60379	61082	15.49	10.60	9.56

表2. 実験結果 経路活性化困難故障

回路名	代表故障数	対象故障数(ランダムパターン検出困難故障:検出100回未満)	バックトラックリミット0回				バックトラックリミット5回				バックトラックリミット10回			
			検出	冗長	打ち切り	CPU時間(sec)	検出	冗長	打ち切り	CPU時間(sec)	検出	冗長	打ち切り	CPU時間(sec)
b14	22802	19371	18678	143	550	395.83	19159	147	65	403.99	19159	148	64	403.81
b15	21988	15407	11181	390	3835	886.16	14114	435	857	1000.58	14226	452	728	1033.98
b20	45459	38407	37723	284	400	850.38	37977	305	125	854.10	37981	305	121	855.71
b21	46154	38970	38130	342	498	920.45	38511	367	92	925.75	38516	367	87	923.84
b22	67536	57074	55833	243	998	1358.20	56383	319	372	1369.74	56424	319	331	1375.03

表3. 実験結果 ATPG困難故障

回路名	代表故障数	対象故障数(経路活性化困難故障:バックトラックリミット5回打ち切り)	Conflictsリミット100回			Conflictsリミット500回			Conflictsリミット1000回			Conflictsリミット無し		
			検出	冗長	打ち切り	検出	冗長	打ち切り	検出	冗長	打ち切り	検出	冗長	打ち切り
b14	22802	65	20	4	41	50	5	10	51	6	8	56	9	0
b15	21988	857	141	22	694	364	48	445	426	63	368	565	292	0
b20	45459	125	21	9	95	92	9	24	104	9	12	111	14	0
b21	46154	92	13	5	74	64	7	21	75	7	10	81	11	0
b22	67536	372	80	15	277	300	17	55	337	17	18	347	25	0