ISSN 2186-5647

2-71

静的学習における並列含意操作グループ化評価

日大生産工（院）　○秋山　正碩　　日大生産工（院）　山崎　紘史
日大生産工　　細川　利典　　九大（院）　　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)が大規模化し，テスト生成時間の増加が問題となっている[1][2].

テスト生成を効率化する技術の一つとして，SOCRATES[3]で提案された静的学習[3]がある．テスト生成および静的学習の処理において含意操作[2]を行っており，その処理時間はテスト生成時間の大半を占める．含意操作とは回路内の信号値を一意に決定する処理である．含意操作は直接含意[4][5]と間接含意[3]に分けられる．静的学習によりテスト生成の前処理で間接含意を求め，テスト生成中の含意操作で使用することで，テスト生成時間を大幅に削減することができる[3]．また，静的学習を繰り返し実行することで，得られる間接含意の数が増加し，テスト生成時間をさらに高速化することが可能である．

しかしながら，近年 LSI の大規模化に伴い，静的学習の処理時間の増加が問題となり，静的学習の高速化が必要となる．本論文では，静的学習を高速化するための技術として，含意操作を並列に処理する並列含意操作[4][6]を用いた静的学習および，並列含意操作を行うためのグループ化法を提案する．

本論文では静的学習により得られた間接含意を，充足可能性問題(Satisfiability problem ： SAT)[7]に基づいたテスト生成に利用し，テスト生成時間を評価する．

2.含意操作

含意操作はゲートの入出力の接続関係から値を求める直接含意[2]と，ゲートの入出力だけでは値を求めることのできない間接含意[2]がある．また直接含意は前方含意と後方含意に分けられる．

図 1 に 2 入力 AND ゲートに対する直接含意の例を示す．図 1(a)は前方含意の例である．信号線 A の値が 0 のとき，AND ゲートの一方の入力に 0 が割当てられたとき出力が 0 となるため，信号線 C の値が 0 となる．このように信号線の入力側から出力側に値を決定する処理が前方含意である．図 1(b)は後方含意の例である．信号線 C の値が 1 のとき，AND ゲートの出力が 1 となるのはすべての入力が 1 でなくてはならないため，信号線 A,B の値は 1 となる．このように出力側から入力側に値を決定する処理が後方含意である．

図 2 に間接含意例を示す．信号線 F の値が 1 のとき，信号線 D,E のいずれかが 1 となる．信号線 D が 1 となるには信号線 A,B が 1 となる．また信号線 E が 1 となるには信号線 B,C が 1 となる．このとき信号線 D,E が 1 となるために，信号線 B が 1 となる必要がある．これより信号線 F の値が 1 ならば信号線 B の値が 1 と決定できる．このように直接含意だけで決定できない含意操作が間接含意である．

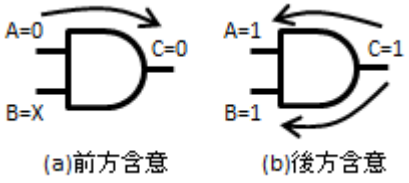

図 1. 直接含意の例

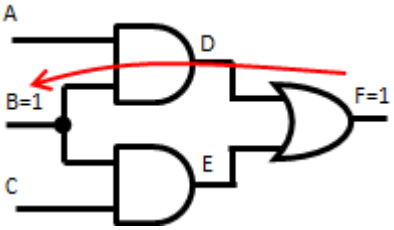

図 2. 間接含意の例

An Evaluation of Grouping Methods for Parallel Implication on Static Learning

Masahiro AKIYAMA, Hiroshi YAMAZAKI, Toshinori HOSOKAWA, and Masayoshi Yoshimura

3.静的学習

静的学習とは直接含意の結果から特定の割当てを求める処理で，繰返し実行することでより多くの間接含意を得られる．本論文における静的学習では対偶規則と固定値学習の 2 つを用いる．

3.1.対偶規則

対偶規則とはある命題が真のとき，対偶も真となる規則である．図 3 に対偶規則の例を示す．信号線 B の値に 0 を割当てたとき，前方含意の結果信号線 D の値が 0 となる．これより “B=0 ならば D=0”という命題が真になる．文献[3]で提案された学習基準に基づいて，この命題に対偶規則を適用すると “D=1 ならば B=1”という命題が真となるため，この間接含意を得ることができる．

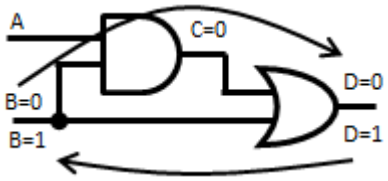

図 3. 対偶規則の例

3.2.固定値学習

固定値学習とは含意操作中に衝突が発生したとき，論理値が 0 または 1 に固定される信号線の値を学習する処理である．図 4 に固定値学習の例を示す．信号線 C の値に 0 を割当て後方含意すると，信号線 A,B の値が 0 となる．さらに信号線 B を後方含意すると，信号線 A の値が 1 となる．このとき信号線 A で値 0 と値 1 が衝突するため，信号線 C に 0 を割当てることはできず，信号線 C の値は 1 と固定値学習できる．

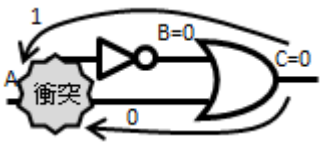

図 4. 固定値学習の例

4.並列含意操作とグループ化

並列含意操作は 2 章で説明した含意操作を，複数同時に処理する操作である．この操作により静的学習における含意操作にかかる処理時間を削減することができる．

図 5 に並列含意操作の例を示す．図 5 において信号線 A,B,C,D に 0 を割当てた場合のそれぞれの含意操作について考える．このとき信号線 A に 0 を割当てた場合，信号線 E,G に 2 回含意操作が行われる．同様に信号線 B,C,D に割当てた場合も 2 回含意操作が行われる．このように並列含意操作を用いない場合は合計 8 回含意操作が行われる．次に並列含意操作を考える．このとき信号線 A の 0 割当てと信号線 B の 0 割当ては同じ NAND ゲートを使い信号線 E に含意操作されるため，並列含意の場合信号線 A,B から信号線 E の含意操作が 1 回となる．同様に信号線 C,D から信号線 F の含意操作も 1 回となる．次に信号線 E,F から信号線 G の含意操作も同じゲートにより含意操作されるため，信号線 E,F から信号線 G の含意操作が 1 回となる．これより並列含意操作を行った場合含意操作の回数は 3 回となり，並列含意操作を行わない場合と比べ含意操作の回数を 5 回削減できる．

並列含意操作の効果を最大限の生かすためには，同時に含意する信号線のグループ化が非常に重要である．図 5 の例では，A=0，B=0，C=0，D=0 の含意操作を同じグループにして，実行することが重要である．グループ化法として，含意操作のイベントが発生するゲートができる限り同じである含意をグループ化するアルゴリズムが効果的である．それゆえ，回路をファンアウトフリー領域(Fanout Free Region ： FFR)に分割し，外部入力から FFR を深さ優先アルゴリズムを用いてグループ化する方法を提案する．

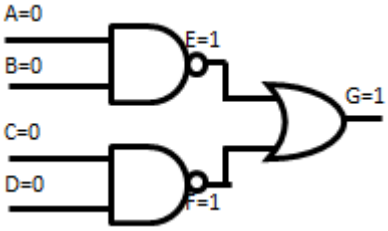

図 5. 並列含意操作例

5.SAT を用いたテスト生成

本論文では静的学習の評価に SAT を用いている．SAT とは乗法標準形(Conjunctive Normal Form: CNF)が与えられたとき，すべての変数の値を 1(真)

または 0(偽)のどちらかに定めることで，式全体の値を 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するとき充足可能(SAT)といい，与えられた CNF が正当化可能であることを示す. SAT は CNF を入力として問題を解決するため，SAT を用いてテスト生成を行うためには，論理回路を CNF に変換する必要がある.

5.1.学習情報の CNF 変換

本論文では静的学習により得た間接含意および固定値情報を SAT に追加するため，これらの情報を CNF 式に変換する必要がある．図 6 に学習した情報を CNF 式に変換する例を示す.

図 6(a)は“F=1 ならば B=1”という間接含意の例である．このとき F=1 のとき B の値は 1 となるが，F=0 のときや B=1 のときはもう一方の値が決定しないため，CNF 式は(-F + B)となる．図 6(b)は“C=1”という固定値の例である．このとき C の値は必ず 1 であるという情報を加えるだけでいいため，CNF 式は(C)となる．このように学習情報を変換した CNF 式を SAT で解く CNF 式に追加することで，テスト生成時間の削減を行う.

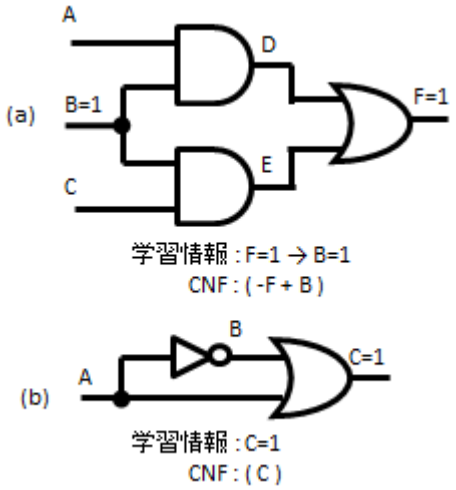

図 6.学習情報の CNF 変換例

6.実験結果

本論文では並列含意操作を用いた静的学習を実装し，間接含意情報，固定値情報を得た．また，固定値情報を SAT に導入し，ISCAS’85 ベンチマーク回路に対しテスト生成の評価を行った.

表 1 に各回路において静的学習で得られた間接含意数と固定値数を示す．表の左から順に回路名，間接含意数，固定値数を示している.

表 2-1 に学習情報を用いないときの SAT によるテスト生成の結果を示す．表の左から順に回路名，対象故障数，冗長故障数，故障検出率，テスト生成にかかった時間を示している.

表 2-2 に固定値学習情報を用いたときの SAT によるテスト生成の結果を示す．各項目は表 2-1 と同じである．回路により学習なしの場合より実行時間が削減できたものと，増加したものに分かれた.

表 2-3 に SAT における学習の有無による時間の比較表を示す. 表の左から順に回路名, 学習なしの CNF 変換にかかった時間，学習なしの SAT にかかった時間，学習有りの CNF 変換にかかった時間，学習有りの SAT にかかった時間である．この結果から実行時間が削減できた回路では CNF 変換, SAT ともに実行時間を削減できていることがわかる.

表 1.静的学習情報数

回路名	間接含意数	固定値数	回路名	間接含意数	固定値数
c17	1	0	c3540	6703	1
c880	124	0	c5315	3245	1
c1355	304	0	c6288	1178	17
c1908	1323	0	c7552	10964	4
c2670	2015	10			

表 2-1.学習無しの SAT による実験結果

回路名	対象故障数	冗長故障数	故障検出率	テスト生成時間
c2670	3630	112	95.74	92.76
c3540	3428	137	96.00	715.02
c5315	5350	59	98.88	266.31
c6288	7744	34	99.30	940.99
c7552	7550	131	98.23	410.25

表 2-2. 固定値学習有りの SAT による実験結果

回路名	対象故障数	冗長故障数	故障検出率	テスト生成時間
c2670	3630	112	95.74	93.08
c3540	3428	137	95.27	712.87
c5315	5350	59	98.84	266.36
c6288	7744	34	99.38	925.44
c7552	7550	131	98.24	402.12

表 2-3.SAT における時間比較

回路名	学習なし		学習有り	
	CNF	SAT	CNF	SAT
c2670	6.11	86.66	6.27	86.73
c3540	12.70	702.33	12.80	699.71
c5315	16.38	249.93	16.39	249.81
c6288	52.78	888.21	51.83	872.44
c7552	34.75	375.50	34.04	367.77

7.おわりに

本論文では並列含意操作を用いた静的学習を実装し，得られた学習情報を用いた SAT によるテスト生成を評価した．実験結果では固定値学習の情報を追加することによりテスト生成時間が増加したものと削減できたものに分かれたが，全体的に大きな変化は生じなかった．今後は固定値に比べ数が多い間接含意を SAT に組込み，テスト生成時間の変化を評価する．

「参考文献」

[1] H. Fujiwara, “Logic Testing and Design for Testability”, The MIT Press (1985)

[2] M. Abramovici, M. A. Breuer and A. D. Friedman, “Digital systems testing and testable design”, IEEE Press, (1995)

[3] MICHAEL H. SCHULZ, ERWIN TRISCHELER “ SOCRATES：A Highly Efficient Automatic Test Generation System”, IEEE TRANSACTION COMPUTER-AIDED DESIGN, VOL 7, NO1, JANUARY (1988) p130

[4] 南山哲郎　高松雄三 “組合せ回路における冗長故障の判定法に関する考察” 電子情報通信学会論文誌 (1998) pp1149-1156

[5] 梶原誠司　猿渡慶太郎 “静的学習における効率的な間接含意の発見と保存について” 電子情報通信学会　TECHNICAL REPORT OF IEICE (2002) pp25-28

[6] Mahesh A, Iyer and Miron Abramovici “FIRE; A Fault-Independent Combinational Redundancy Identification Algorithm”, IEEE TRANSACTIONS ON VERY LARGESCALE INTEGRATION (VLSI) SYSTEMS, VOL 4, NO 2, (1996), pp295-301

[7] Tracy Larrabe., “Test Pattern Generation Using Boolean Satisfiabillity”, IEEE TRANSACTION ON COMPUTER-AIDED DESIGN, VOL 11, NO 1, (1992)