ISSN 2186-5647

相対時刻静的学習によるテスト生成の効率化

日大生産工（院）　○小松　正樹　日大生産工　細川　利典
九大　吉村　正義　　明大　山崎　浩二

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 Cricuits: LSI）が大規模化し，テスト生成時間[1][2]の増加が問題となっている．

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][4]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[5][6]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計算し，テスト生成中の含意操作で利用することで，テスト生成時間を削減する[3]．

静的学習の利点として，テスト生成中の解の探索空間が削減されることが挙げられるが，静的学習自体の実行時間の増加が問題となっている．特に順序回路は時間展開[1]を用いて静的学習を展開時間数分繰り返すため，展開時間数の増加による静的学習の増加が問題となっている．本論文では，相対時刻静的学習を用いて，順序回路に対する静的学習を高速化する．「時刻 s の信号線 p が決定すると，時刻 t の信号線 q が決定する」と絶対的に時刻と信号線で位置指定をした静的学習を，「信号線 p が決定すると，p から n 時刻離れた信号線 q が決定する」と相対的に位置指定をするアルゴリズムに変更することで静的学習を高速化する．

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ことのできる直接含意[5][6]と，ゲートの入出力の接続関係だけでは値を求めることのできない間接含意[3]がある．

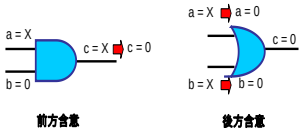

図 1　直接含意の例

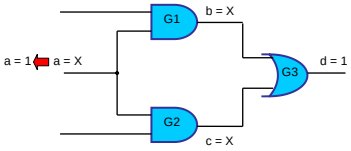

図 2　間接含意の例

また直接含意は前方含意と後方含意に分けられる．

図 1 に直接含意例を示す．図 1 の左例は前方含意の例である．信号線 b の信号値が 0 である場合，AND ゲートの接続関係から信号線 c の信号値は 0 と決定できる．このようにゲートの入力から出力側に値を決定する処理が前方含意である．図 1 の右例は後方含意の例である．信号線 c の信号値が 0 である場合，OR ゲートの接続関係から，信号線 a，b の信号値は 0 と決定できる．このように出力から入力側に値を決定する処理が後方含意である．直接含意は論理ゲートの入出力関係と推移律を利用して容易に得ることができる含意である．

図 2 に間接含意例を示す．信号線 d の信号値を 1 とした場合，直接含意では他の信号線の値は決定できないが，b=1 または c=1 のいずれかが成り立たなければならないため，いずれの場合も a=1 が必要であることがわかる．図 2 のように“d=1 ならば a=1”と決定する処理が間接含意である．

3. 時間展開モデル

k 時間展開モデルとは，1 時刻目のスキャン FF の出力を擬似外部入力とし，k 時刻目のスキャン FF のデータ入力を擬似外部出力として，k 時間分順序回路を時間展開した組合せ回路モデルである．

図 3 に k 時間展開モデルの例を示す．各回路 C(i)は時刻 i に対応する順序回路の組合せ回路部に対応している．時刻：0 のセル C(0)の FF の出力を疑似外部入力，時刻：k-1 のセル C(k-1)の FF の入力を擬似外部出力と呼ぶ．

On the Efficiency of Test Generation Using Relative Time Static Learning

Masaki KOMATSU, Toshinori HOSOKAWA, Masayoshi YOSHIMURA and Koji YAMAZAKI

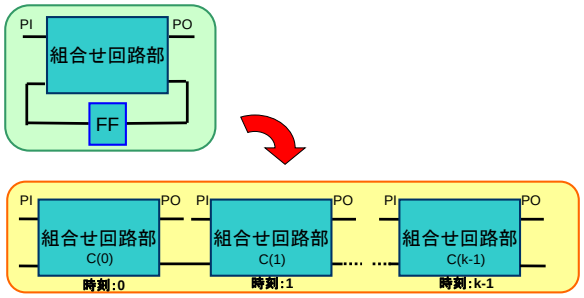

図 3　k 時間展開モデルの例

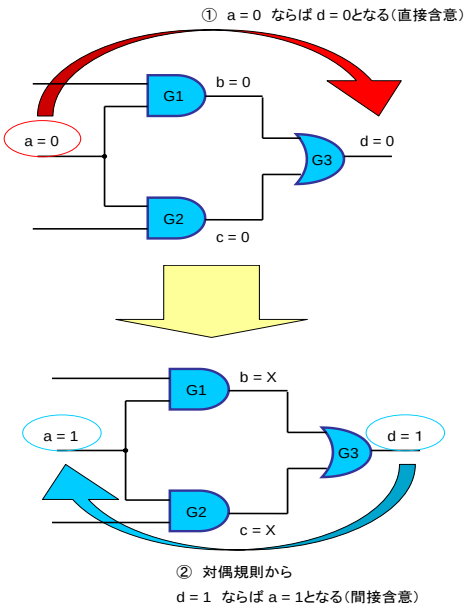

図 4　静的学習の例

順序回路の静的学習は任意の時間展開した時間展開モデルを用いて，時刻を越えて間接含意を計算する順序回路の静的学習を行う．

4. 静的学習

静的学習とは直接含意の結果から間接含意を求める．そのため命題が真ならばその対偶も真であるという対偶規則を用いる．

図 4 に静的学習の例を示す．まず信号線 a の信号値を 0 とし，前方含意操作を行なう．その結果，“a=0 ならば d=0”とわかる．この対偶“d=1 ならば a=1”も真となるため“d=1 ならば a=1”という間接含意を学習することができる．

以下に示す学習規則 A または B[3]を満たす直接含意が存在するとき間接含意を保存する．

[学習規則 A]

信号線 s の値を v(v∈{0, 1})とする含意操作により，ある 2 入力以上のゲートの出力信号線 t が値 w(w∈{0, 1})をとり，かつ，w がそのゲートの非制御値であるとき，間接含意として”t = ¬w ならば s = ¬v”を保存する．

ここで非制御値は AND ゲートに対する論理値 1 のようにゲートの入力値がすべて決定しているときに出力も決定する値のことである．

[学習規則 B]

信号線 s の値を v(v∈{0, 1})とする含意操作において，後方への含意操作により，ある 2 入力以上のゲート入力信号線 t が値 w(w∈{0, 1})となり，かつ，w がそのゲートの制御値であるとき，間接含意として”t = ¬w ならば s = ¬v”を保存する．

学習を実行する信号線の順序は文献[7]で学習順序として効果が高いとされている回路の論理段数を基にした前方への幅優先探索による学習順序を用いる．

5. 絶対時刻静的学習

絶対時刻静的学習では時間展開モデルのすべての時刻に対して 4 章の静的学習を行う．図 5 に絶対時刻静的学習のアルゴリズムを示す．絶対時刻静的学習では展開時間数の増加にともない静的学習の実行時間が増加する．

以下に各ステップの処理内容を説明する．

(Step1)

すべての信号線に対して絶対時刻静的学習を繰り返す．

(Step2)

すべての時刻に対して絶対時刻静的学習を繰り返す．

(Step3)

任意の信号線と時刻から絶対時刻静的学習を行う．

(Step4)

絶対時刻静的学習を行う時刻を変更して(Step2)に戻る．

(Step5)

絶対時刻静的学習を行う信号線を変更して(Step1)に戻る．

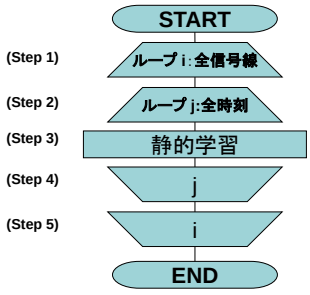

図 5　絶対時刻静的学習のアルゴリズム

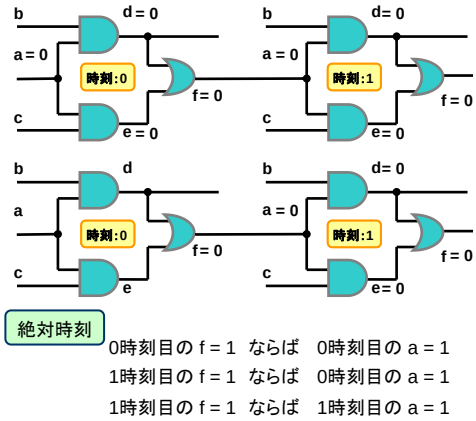

図 6　絶対時刻静的学習の例

図 6 に 2 時間展開の絶対時刻静的学習の例を示す．まず 0 時刻目の信号線 a を論理値 0 としての静的学習から「0 時刻目の f=1 ならば 0 時刻目の a=1」「1 時刻目の f=1 ならば 0 時刻目の a=1」という間接含意を得る．次に 1 時刻目の信号線 a を論理値 0 としての静的学習から「1 時刻目の f=1 ならば 1 時刻目の a=1」という間接含意を得る．

6. 相対時刻静的学習

図 7 に相対時刻静的学習における時間展開の例を示す．相対時刻静的学習では 2 時間展開をする場合，中心時刻より前後に 2 時間分の時刻展開を行う．そのため 3 時間展開の回路を使用する．相対時刻静的学習ではこのように時間展開した回路の中心の 0 時刻目からのみ静的学習を行う．

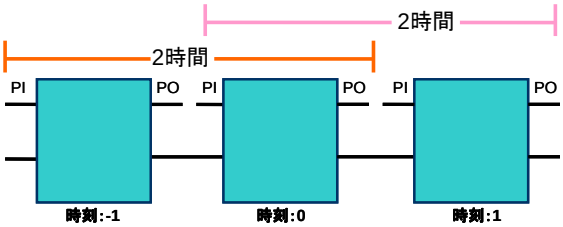

図 7　相対時刻静的学習の時間展開

図 8 に相対時刻静的学習のアルゴリズムを示す．相対時刻静的学習では中心の時刻からのみ静的学習を行うため，時刻分の繰り返しを必要としない．そのため絶対時刻静的学習に比べ，実行時間が削減される可能性がある．

以降で各ステップの処理内容を説明する．

(Step1)

すべての信号線に対して相対時刻静的学習を繰り返す．

(Step2)

任意の信号線から相対時刻静的学習を行う．

(Step3)

相対時刻静的学習を行う信号線を変更して(Step1)に戻る．

図 9 に 2 時間展開の相対時刻静的学習の例を示す．相対時刻静的学習では 3 時間分に時間展開された回路を用いて，中心の 0 時刻目から静的学習を行う．そして 0 時刻目の信号線 a を論理値 0 とした含意から「f=1 ならば 0 時刻後の a=1」「f=1 ならば 1 時刻後の a=1」という間接含意を得る．この間接含意は図 6 で得られる 3 つの間接含意と同じ意味である．そのため相対時刻静的学習では間接含意を保存するメモリ容量も削減できる可能性がある．

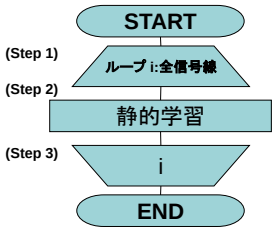

図 8　相対時刻静的学習のアルゴリズム

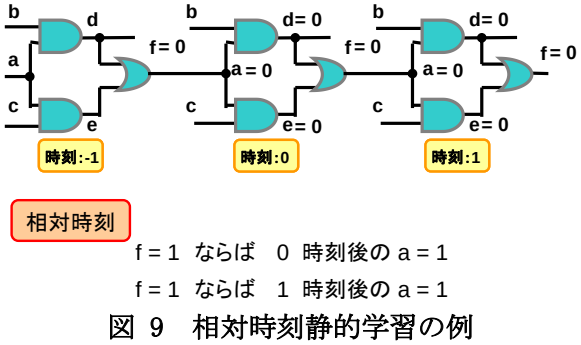

図 9　相対時刻静的学習の例

表 1　相対・絶対時刻の間接含意数の比較

	間接含意数(規則A)		間接含意数(規則B)	
回路	絶対	相対	絶対	相対
s1488	25675	10563	1146	450
s1494	26204	10788	1220	499
s5378	47078	14679	3404	2054
s9234	165254	27286	8948	910
s35932	36819	19740	13026	4918

表 2　相対・絶対時刻の実行時間の比較

	実行時間(sec)	
回路	絶対	相対
s1488	1.61	0.67
s1494	1.61	0.74
s5378	8.83	2.12
s9234	23.61	4.87
s35932	208.14	93.1

7. 実験結果

相対時刻静的学習を実装し実験を行った. 実験は ISCAS'89 ベンチマーク順序回路に対して行い, 展開時間は 3 時間とする.

表 1 は相対時刻静的学習と絶対時刻静的学習の間接含意数を比較したものである. 間接含意数 (規則 A) は学習規則 A で得られた間接含意数を, 間接含意数 (規則 B) は学習規則 B で得られた間接含意数を示す. 実験の結果, 相対時刻静的学習は学習規則 A で 46～83%, 学習規則 B で 39～89%の間接含意数を削減した.

表 2 は相対時刻静的学習と絶対時刻静的学習の実行時間を比較したものである. 実験の結果, 相対時刻静的学習は 54～79%実行時間を削減した.

実験結果から, 相対時刻静的学習は, 絶対時刻静的学習に比べ, メモリ容量・実行時間ともに削減できる有効な手法であることがわかった.

8. おわりに

本論文では, 順序回路の静的学習の実行時間とメモリ容量を削減するために相対時刻静的学習を提案し, 実装した. そして絶対時刻静的学習との比較を行った結果, 絶対時刻静的学習に比べ, メモリ容量・実行時間を削減し, 有効な手法であることがわかった. 今後は含意操作の並列化など他の, アプローチからの静的学習の高速化手法を提案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] M. Abramovici, M. A. Breuer and A. D. Friedman, "Digital systems testing and testable design," IEEE Press, (1995).

[3] MICHAEL H. SCHULZ, ERWIN TRISCHLER "SOCRATES : A Highly Efficient Automatic Test Generation System" IEE TRANSACTIONS ON COMPUTER- AIDED DESIGN, VOL 7 NO 1, JANUARY(1988) p130

[4] Jian-Kun Zhao, Jeffrey A. Newquist and Janak H. Patel "A Graph Traversal Based Framework for Sequential logic Implication with an Application to C-cycle Redundancy Identification" IEEE (2000)

[5] 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)

[6] 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)

[7] Hideyuki ICHIHARA, Seiji KAJIHARA, Kozo KINOSITA : On Processing Order for Obtaining Implication Relations in Static Learning , IEICE Trans. (2000).