

C-015

必須割当情報に基づくテストデータ削減のためのテストポイント挿入法 A Test Point Insertion Method for Test Data Reduction Based on Necessary Assignment

平本 和子¹, 吉川 祐樹¹, 市原 英行¹, 井上 智生¹
Kazuko Hiramoto, Yuki Yoshikawa, Hideyuki Ichihara, Tomoo Inoue

あらまし

LSIの大規模化により、テスト実行時間の増加が重要な問題となっている。本論文では、必須割当の衝突が原因で同じテストパターンでテストできない2つの故障の衝突は、テストポイントを用いて解消可能であることに着目し、故障の衝突解消度の高い信号線にテストポイントを挿入することでテストデータ量を削減させるアルゴリズムを提案する。ベンチマーク回路に対する実験結果では、提案するアルゴリズムが適用した全ての回路に対するテストパターンを削減でき、一部の回路に対しては効果的にテストデータ量も削減できることを示した。

1 背景

近年の半導体技術の進展に伴い、LSIの回路規模が増大し、テスト実行時間の増加が問題となっている。自動テストパターン生成ツール(ATPG)によって高い故障検出率を得るテストパターンを生成するには、フルスキャン設計方法に代表されるテスト容易化設計(DFT)が必要である。

DFTの手法の一つとして、テストポイント(TP)挿入によるテストデータ削減法が存在する。文献[1]では、確率的指標によりTP挿入点を決定し、テストパターン圧縮率を向上させることでテストデータを削減する手法が提案されている。

本論文では、2つの故障が同じテストパターンでテストできない原因、つまりテストパターン数が増加する原因に着目したTP挿入法を提案する。テストパターン数増加の原因の1つである故障の必須割当の衝突を解消することで、テストデータ量が削減できる。

2 テストポイントとテストパターン

TPは制御点と観測点の2種類がある[2]。制御点は可制御性、観測点は可観測性を向上させることができる。

文献[1]で示されているTPのように、一般的にはフルスキャン設計を対象とした、マルチプレクサとスキャンFFを用いた制御も観測も可能なTPを使用する。本論文では、制御点と観測点のそれぞれに対するTP挿入の効果について考えるため、図1に示すようなシンプルな構造と機能を持つTPを使用する。図1の制御点と観測点は、それぞれ別に存在するために挿入の自由度が高くハードウェアオーバーヘッドが小さい。さらに、テスト対象となるTPの故障数が少ないという利点をもつ。

観測点は図1(a)のように、外部出力もしくはスキャンFFに接続する信号線である。挿入により、ゲートAの出力である信号線cの値を信号線dの値に依存せずに観

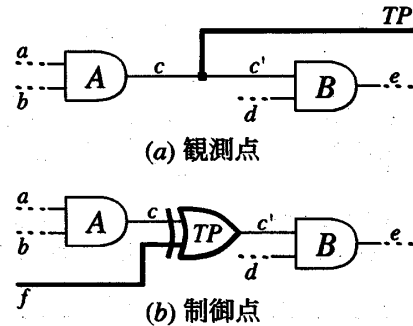


図1: テストポイント

測することができ、Aに到達可能な信号線領域(ファンインコン)の信号線の可観測性を向上させることができる。

制御点は、図1(b)のように、1つの入力を自由に制御できるXORゲートである。信号線fの値を制御することにより、入力値を反転させることができる。例えばf=0のときc'=c、f=1のときc'=c̄となり、cの可観測性を損なうことなくc'の値を制御できる。制御点は、XORゲートの他にANDゲート、ORゲートやマルチプレクサを用いた制御点も存在する。XORゲートを用いた制御点は、論理の反転のみで故障の影響をマスクしないことや、マルチプレクサを使用するよりハードウェアオーバーヘッドが小さい利点がある。

TP挿入技術は主に故障検出率向上のために用いられるが、ここではテストデータ量の削減のために用いる。例えば、図2(a)の信号線aの1縮退故障(a/1)を検出するためのテストパターン(a,b,c)=(0,1,1)と、図2(b)の故障c/1の故障を検出するためのテストパターン(a,b,c)=(0,1,0)は、検出のために割り当てる必要のある信号線cの値が異なるため、同じテストパターンで2つの故障を検出することが出来ない。しかし、図2(c),(d)で示すように、信号線gに制御点を挿入して制御信号線i=1とすることで、a/1を検出するために必要な割当であるc=1が必要なくなり、c/1と同じテストパターン(a,b,c)=(0,1,0)で故障を検出できる。また、a/1とc/1が同じテストパターンで故障を検出可能となる別の方法として、図2(e),(f)で示すように信号線fに観測点を挿入する方法がある。これにより、a/1を検出するために必要な割当であるc=1が必要ではなくなり(ドントケア)、c/1と同じテストパターン(a,b,c)=(0,1,0)で故障を検出できる。

提案するTP挿入法は、2つの故障が同じテストパターンで検出できない原因に着目し、その原因を解消するようにTPを挿入する手法である。この手法により、一つのテストパターンでより多くの故障を検出できるようになりテストデータ量が削減できる。

¹広島市立大学大学院情報科学研究科

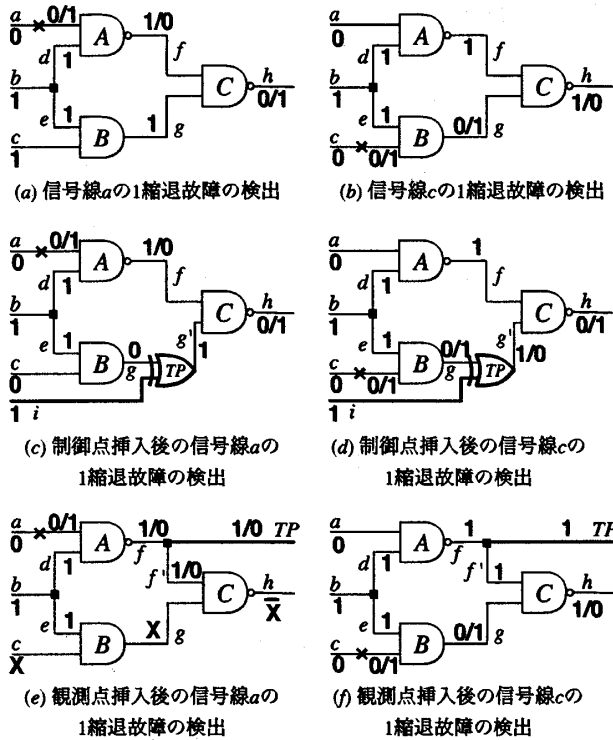


図2: テストポイントによるテストデータ量の削減

3 故障の必須割当に基づく衝突の解消

2つの故障が同じテストパターンで検出できない原因の一つとして、故障の必須割当の衝突がある。本節では、故障の必須割当と必須割当が原因で起こる故障の衝突について述べ、その問題を解消するTPの効果について説明する。

3.1 故障の必須割当情報と衝突

故障を検出するために必要な信号値の割当を必須割当という。図2(a)は、故障a/1を検出する様子を示している(信号値割当は、正常時/故障時を示す)。必須割当は故障を活性化させるために必要な割当と、故障を伝搬させるために必要な割当がある。故障a/1を活性化させるために必要な割当は $a=0$ である。また、故障の影響を伝搬させるために必要な割当は、ゲートAの出力に伝搬させるために必要な $d=1$ とゲートCの出力に伝搬させるために必要な $g=1$ である。さらに、これらの必須割当の含意により決まる $b=1, c=1, e=1, f=1, h=1$ も必須割当である。これらの必須割当により、a/1はテストパターン $(a,b,c)=(0,1,1)$ で故障を検出できる。

2つの故障が同じテストパターンで故障を検出できないことを故障の衝突といい、衝突の原因の一つとして必須割当の衝突がある。図2(b)は、故障c/1を検出するための必須割当を示しており、テストパターン $(a,b,c)=(0,1,0)$ でc/1を検出できる。図2(a)と比較すると、信号線g, h, cで異なる値を必須割当としている。これを故障の必須割当の衝突といい、a/1とc/1が同じテストパターンで故障を検出できない原因の一つであると考えられる。

3.2 制御点による衝突解消

制御点は、信号線の挿入前後の信号値の値を反転することができる。この性質により信号線Lに制御点を挿入することで、信号線 $L=1$ を必須割当とする全ての故障と、信号線 $L=0$ を必須割当とする全ての故障について、Lでの必須割当の衝突を解消できる。

図2(c),(d)は、制御点による故障の衝突解消により、故障a/1と故障c/1が同じテストパターンで故障を検出する様子を示している。a/1は $g=1$ を必須割当とする故障の一つであり、c/1は $g=0$ を必須割当とする故障の一つである。この2つの故障は必須割当が衝突しているが、(c),(d)のように制御点を信号線gに挿入し $i=1$ とすることで、a/1の信号線gの必須割当を0にできるため、同じテストパターンで故障を検出できる。

3.3 観測点による衝突解消

観測点は、挿入によりファンインコーンの信号線の可観測性を高めることができる。ゲートGの入力信号線 $L, M(L \neq M)$ とし、ゲートGの制御値を C_G とすると、信号線Lに観測点を挿入することで伝搬の条件 $M = \overline{C_G}$ を必須とする全ての故障と、 $M = C_G$ を必須とする全ての故障の必須割当の衝突を解消できる。

図2(e),(f)は観測点による故障の必須割当解消により、故障a/1と故障c/1が同じテストパターンで故障を検出する様子を示している。図2(a),(b)より、a/1は信号線fの分岐を含まないファンインコーン領域であるファンアウトフリーリージョン(FFR)に存在する故障であるため、非制御値である $g=1$ を伝搬のための必須割当とする。またc/1は、 $g=0$ を必須とする故障である。この2つの故障の必須割当の衝突は、(e),(f)のように観測点をfに挿入することで、a/1の信号線gの必須割当を不要にして解消できる。この必須割当の衝突の解消により(e),(f)では、2つの故障は同じ1つのテストパターンで故障を検出できる。

制御点と観測点を用いた必須割当の衝突解消例のように、1つのTPにより故障の衝突が解消され、同じテストパターンで故障を検出可能となることがある。一方、複数の必須割当の衝突が原因で互いに衝突する故障は、TPを複数必要とする場合もある。

4 テストデータ削減のためのテストポイント挿入法

1つのTPで解消可能な故障の衝突は複数あり、複数の必須割当の衝突によりTPを複数必要とする場合も存在する。よって、回路全体の故障の衝突をできるだけ多く解消できるTP挿入点を選択することで、少ないTP数でテストデータ量を削減できると考えられる。

本節では、TPでどれだけの故障の衝突を解消できるかを見積もるために、冗長故障判定アルゴリズムのFIRE(Fault Independent Redundancy identification algorithm)[3]で用いられる、信号値割当からそれを必須とする故障を探索する方法を導入し、TP探索アルゴリズムの概要について説明する。

4.1 FIREを用いた故障の必須割当方法

TPで解消可能な故障の衝突を特定するには、必須割当の衝突が原因で衝突する故障を探す必要がある。提案手法は、ある信号値割当からそれを必須とする故障を探索する方法として、冗長故障判定法 FIRE で用いられている故障の探索法を用いる。FIREは、必須割当の矛盾に基づく冗長故障判定アルゴリズムである。

4.1.1 制御点で解消可能な故障衝突

3.2で示したように、信号線 L に制御点を挿入することで $L=0$ を必須とする故障集合 S_0 と、 $L=1$ を必須とする故障集合 S_1 の任意の故障ペア $(f_0, f_1) \in S_0 \times S_1$ の必須割当の衝突を解消できる。

制御点で解消可能な故障の衝突箇所は、FIREで用いられている故障の探索方法を導入することで探索できる。FIREは $L=v$ を含意する必須割当を見つけるため、 $L=\bar{v}$ から含意操作を行う。これは、 $L=\bar{v} \Rightarrow M=u$ が得られると、 $M=\bar{u} \Rightarrow L=v$ なので、 $M=\bar{u}$ が $L=v$ を含意する信号値割当となるからである。また、 $M=\bar{u}$ が故障と直接的な必須割当の関係となる故障は、以下のように一般的にいえる。

直接的な故障と必須割当の関係

1. $L=\bar{v}$ は故障 L/v の必須割当 ($v \in \{0, 1\}$) である。
2. 信号線 L と信号線 M はゲート G の任意の入力 ($L \neq M$) とし、 C_G はゲート G の制御値とする。このとき、 $L=\bar{C}_G$ は信号線 M を出力とする FFR の全ての故障の必須割当である。

前述の考え方を用いて、図3の回路を用いて例を示す。図3(a)に示すように、 $g=0$ を含意する信号値割当を見つけるため、 $g=1$ から含意操作を行う。図3(a)より、 $g=0$ を含意する信号値である $b=0, c=0, d=0, e=0, g=0$ を得る。さらに、故障と直接的な必須割当の関係1より、 $b=1, c=1, d=1, e=1, g=1$ は故障 $b/1, c/1, d/1, e/1, g/1$ の必須割当であることから、 $g=0$ を必須とする故障は、 $b/1, c/1, d/1, e/1, g/1$ となる。同様に図3(b)では、 $g=1$ を必須割当とする信号値割当を、 $g=0$ から含意操作して得ている様子である。これにより、 $g=1$ を必須とする故障は、 $g/0, h/1$ であることがわかる。また、 $g=1$ はゲート C の非制御値であることから、故障と直接的な必須割当の関係2より、 $g=1$ を必須とする故障は $a/1, a/0, d/1, d/0, f/1, f/0$ となる。以上より $g=0$ を必須割当とする故障集合 S_0 と、 $g=1$ を必須割当とする故障集合 S_1 は以下ようになる。

$$S_0 = \{b/1, c/1, d/1, e/1, g/1\}$$

$$S_1 = \{g/0, h/1, a/1, a/0, d/1, d/0, f/1, f/0\}$$

FIREでは、故障 $d/1$ のように S_0 と S_1 の両方に含まれる故障は、不可能な値の組合せを必須としているので冗長故障として見つけることができる。検出できない冗長故障を検出するためのテストパターンは存在しないことから、冗長故障との必須割当の衝突を解消してもテストパターンを削減することができない。よって、 S_0 と S_1

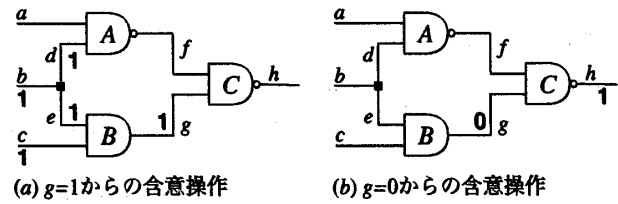


図3: FIREの必須割当判定方法

から冗長故障(上述の例では $d/1$)を除いて、解消可能な故障の衝突数は $|S_0 \times S_1|$ と計算できる。

4.1.2 観測点で解消可能な故障衝突

3.3で示したように、ゲート G の入力信号線 $L, M (L \neq M)$ とし、ゲート G の制御値を C_G とするとき、信号線 L に観測点を挿入することで伝搬の条件 $M=\bar{C}_G$ を必須とする故障集合 $S_{\bar{C}_G}$ と、 $M=C_G$ を必須とする故障集合 S_{C_G} の任意の故障ペア $(f_0, f_1) \in S_{\bar{C}_G} \times S_{C_G}$ の衝突を解消することができる。

図3の回路を用いて例を示すと、信号線 f に観測点を挿入することで解消可能な故障の衝突は、 g に制御値を必須とする故障と、ゲート C に故障の影響を伝搬のために g に非制御値を必須とする故障である。制御値 $g=0$ を必須とする故障は、制御点で示した例と同様に $b/1, c/1, d/1, e/1, g/1$ となる。また、非制御値 $g=1$ を故障の影響を伝搬するために必須とする故障は、故障と直接的な必須割当の関係2より $a/1, a/0, d/1, d/0, f/1, f/0$ となる。以上より、 $S_{C_G}, S_{\bar{C}_G}$ は以下ようになる。

$$S_{C_G} = \{b/1, c/1, d/1, e/1, g/1\}$$

$$S_{\bar{C}_G} = \{a/1, a/0, d/1, d/0, f/1, f/0\}$$

制御点と同様に考え、 S_{C_G} と $S_{\bar{C}_G}$ から冗長故障(上述の例では $d/1$)を除いて、解消可能な故障の衝突数は $|S_{C_G} \times S_{\bar{C}_G}|$ と計算できる。

4.2 テストポイント探索アルゴリズム

提案するアルゴリズムは、入出力以外の各信号線 L の中で、TPで解消可能な故障の衝突数が最も多い信号線とTPの種類を探索する戦略である。挿入する信号線やTPの種類によって衝突解消の効果が異なることから、より多く全体の故障の必須割当の衝突を解消するTPを選ぶことが、テストデータ削減に有効であると考えられる。

制御点および観測点それぞれにおいて解消可能な衝突数 $cnt = |S_0 \times S_1|$ と $ob = |S_{C_G} \times S_{\bar{C}_G}|$ を計算し、 cnt と ob を正規化して衝突解消可能な信号線とTPの種類を探索する。正規化は、制御点と観測点のどちらかに偏りが生じないようにするために行う。また、正規化した cnt には α 、 ob には β をそれぞれ重み付けし、制御点と観測点の衝突解消によるテストデータ削減の効果を調整可能とする。

ここで提案するTP探索アルゴリズムは、以下に示すように回路情報とTP挿入数 N を入力とし、TP挿入信号線と挿入するTPの種類(制御点または観測点)を N 個得る方法である。

テストポイント探索アルゴリズム

1. 入出力以外の各信号線 L に対して,
 - (a) 信号線 L に制御点を挿入することで解消可能な故障の衝突数 $cnt = |S_0 \times S_1|$ を求める.
 - (b) 信号線 L に観測点を挿入することで解消可能な故障の衝突数 $ob = |S_{CG} \times S_{\bar{C}G}|$ を求める.
2. cnt と ob の正規化を行い, 重み α と β で $cnt \cdot \alpha$ と $ob \cdot \beta$ のように重み付けする.
3. 2 で求めた $cnt \cdot \alpha$ と $ob \cdot \beta$ が最も大きいものを N 個取り出す.

5 実験結果

提案する TP 挿入点探索アルゴリズムを用いて実験を行った. TP 挿入数は, 外部入力数 (PI), 外部出力数 (PO), フリップフロップ数 (FF) の総和の 10% とし, TP 挿入点探索アルゴリズムの中で用いる正規化後の重み付け α と β は $\alpha + \beta = 1$ となるように 0~1 の範囲で 0.25 刻みで実験を行った. 9 つの ITC99 ベンチマーク回路を対象とし, TP 探索の計算は Apple PowerPC G5 (CPU: Dual 2GHz) 上で行い, テスト生成には TetraMAX (Synopsis) を用いた. 今回の実験で評価したテストデータ量 (TD) は, 以下の式 1 で求めた.

$$TD = (FF + TP) \times (\text{テストパターン数}) \quad (1)$$

ここで TP は挿入した TP 数を意味する. 実験では回路がフルスキャン設計されていると仮定しており, 式 1 では単純化のために, 1 つの観測点または制御点には, 1 つの FF が必要になると考えている.

表 1 に TP 挿入前 (org) と, すべてのパラメータ対の中で最も結果が良かった $(\alpha, \beta) = (0.25, 0.75)$ を用いて提案する TP 探索法を用いた場合 (prop) と, ランダムに TP を選択した方法 (rand) のテスト生成後のテストパターン数とテストデータ量の結果を示す. 最後の列 (time) には, prop の TP 探索と TP 挿入にかかった計算時間を示す.

表 1 に示すように提案法は, TP 挿入前やランダムと比較して, 全ての回路のテストパターン数を削減できていることがわかる. テストデータ量では, 提案法はランダムに比べて少なく, さらに b04, b09, b14 の回路はテストデータ量を削減できた. 特に b09 では, もとのテストデータ量に比べて 25.4% 程度の削減ができた. 一方, 他の回路ではテストポイント挿入前のほうがデータ量が少なかった. これは式 1 に示すように, テストパターンが削減できたことによるデータ量削減よりも, TP による FF の増加によるテストパターン長の増加が大きかったためである.

TP 挿入点探索の計算時間は, 信号線数の 2 乗にほぼ比例して増加している. これは, 今回の実験では入出力信号線を除く全ての信号線で TP 探索を行ったためであり, 実用的な計算時間でテストポイント探索を行うためには, ヒューリスティックにより探索を限定する必要があると考えられる.

表 2 は, 提案法を用いた b09 の重み (α, β) の違いによるテストパターン数とテストデータ量の結果である.

表 1: $(\alpha, \beta) = (0.25, 0.75)$ の時の 10% の TP 挿入によるテストパターン数とテストデータ量と計算時間

回路	テストパターン数			テストデータ量 TD [bit]			time [s]
	org	prop	rand	org	prop	rand	
b04	96	87	100	6438	6391	7200	17
b07	47	42	46	2000	2072	2160	6
b08	38	33	41	1080	1088	1184	1
b09	35	24	36	928	672	992	1
b10	46	44	49	1148	1216	1271	2
b11	87	83	88	3154	3311	3444	40
b13	37	35	38	2701	2835	3078	91
b14	1015	915	936	335965	333957	341640	58266
b15	675	641	688	374625	391651	420368	88809

表 2: 提案法を用いた b09 の (α, β) の違いによるテストパターン数とテストデータ量

(α, β)	テストパターン数	TD [bit]	削減率 [%]
org	35	1190	-
(1, 0)	37	1369	+15.0
(0.75, 0.25)	36	1332	+11.9
(0.5, 0.5)	24	888	-25.4
(0.25, 0.75)	24	888	-25.4
(0, 1)	24	888	-25.4

削減率 [%] はもとのテストデータ量に対する増減率を表している. テストパターン数もテストデータ量も観測点の重み β が高い方が, 大幅に削減できていることがわかる. よって, 実験データで $(\alpha, \beta) = (0.25, 0.75)$ の結果がいずれの回路でもテストパターン数削減の効果があったのは, 制御点より観測点挿入による故障の衝突解消がテストパターン数削減に効果的であるためと考えられる.

6 まとめ

本論文では, 故障の衝突原因の 1 つである必須割当の衝突を TP を用いて解消可能であることに着目し, 故障の衝突解消度の高い信号線に TP を挿入することでテストデータ量を削減させるアルゴリズムを提案した. 実験結果より, 全ての回路のテストパターン数が削減でき, 一部回路でテストデータ量を大幅に削減できたことを示した.

今後の課題は今回の実験結果を踏まえて, よりいっそうのテストデータの削減や, テストポイント探索の計算時間削減を考慮したヒューリスティックの提案, 従来法との比較による有効性の確認を行う.

参考文献

- [1] 吉村, 細川, 太田, "ATPG パターン数削減指向テストポイント挿入方法", 電子情報通信学会論文誌, Vol. J86-D-I, No. 12, pp. 884-896, 2003.
- [2] N. K. Jha, S. Guputa, Testing of Digital System, Cambridge University Press, 2003.
- [3] M. A. Iyer and M. Abramovici, "Low-Cost Redundancy Identification for Combinational Circuits," 7th International Conference on VLSI Design, pp.315-318, Janary. 1994.