

ドントケア抽出技術を用いたブリッジ故障検出の効率化に関する研究

日大生産工（学部）○若園 大洋 日大生産工 細川 利典

1. はじめに

近年, VLSI の大規模化, 高速化, 複雑化に伴い, 縮退故障のテストだけでは不十分であり他の故障モデルでのテストが必要となっている. 2) 縮退故障以外の代表的な論理故障モデルとしてブリッジ故障があり, 実際に VLSI で頻繁に発生している信号線間の短絡をモデル化したものである. しかしながら, 縮退故障モデル以外の故障モデルでのテストにおいては, テスト生成時間とテストパターン数が増加し, テストコストが増加してしまう. 高品質なテスト手法として n 回検出テストに対してドントケア抽出を行い, ドントケアにブリッジ故障を検出するような値を再割り当てることにより, ブリッジ故障検出率を高める手法 3) ある. この手法でのブリッジ故障検出条件は AND 型か OR 型どちらかを検出した場合である. しかし n 回検出では高品質の故障検出が期待できるが, その分テストパターン数が多くなってしまふ. 本稿では, より少ないテストパターンでテストを行うために, 1 回検出の縮退故障用のテストパターンを用いてブリッジ故障を検出する方法を検討する. さらに検出条件が厳しい U モデルを採用し, ブリッジ故障検出の品質を高める. ドントケア抽出技術を用いて縮退故障用のテストパターン中のドントケアを抽出し, ブリッジ故障検出率ができるだけ高くなるドントケアの再割り当てを提案する.

2. ブリッジ故障の検出

ブリッジ故障とは, 回路にほこり等が付着しある二つの信号線が短絡する欠陥をモデル化した故障モデルである. ブリッジ故障には AND 型ブリッジ故障と OR 型ブリッジ故障の二つのタイプがある. AND 型ブリッジ故障とは二つの信号線の間にあたかも AND ゲートがあるかのようにふるまふ. OR 型ブリッジ故障とは二つの信号線の間

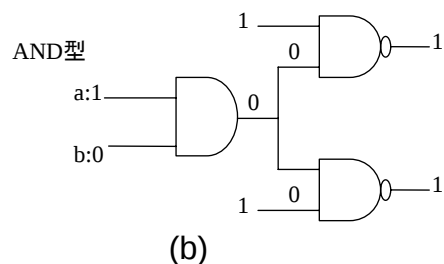
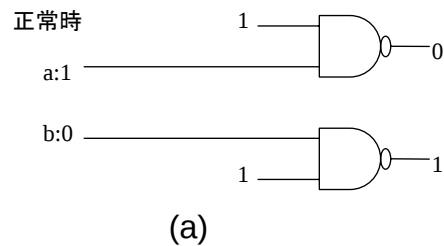
あたかも OR ゲートがあるかのようにふるまふ.

本稿では, ブリッジ故障はノンフィードバックブリッジ故障のみを対象とする. ノンフィードバックブリッジ故障とは, ブリッジが起ったときにフィードバックを起こさないブリッジ故障である.

AND 型ブリッジ故障を検出するには, 信号線 $a(b)$ の 0 縮退故障を検出し, かつ信号線 $b(a)$ の論理値を 0 とするテストパターンを生成すればよい (図 1 (b)).

OR 型のブリッジ故障を検出するには信号線 $a(b)$ の 1 縮退故障を検出し, かつ信号線 $b(a)$ の論理値を 1 とするテストパターンを生成すればよい (図 1 (c)).

本稿ではブリッジ故障モデルとして U モデルを採用する. U モデルの検出は, AND 型のブリッジ故障と OR 型のブリッジ故障の両方が検出できた場合のみ, 二つの信号線間のブリッジが検出できたとする.



Efficiency of Detecting Bridging Fault Using Don't Care Identification Technique

Motohiro WAKAZONO, and Toshinori HOSOKAWA

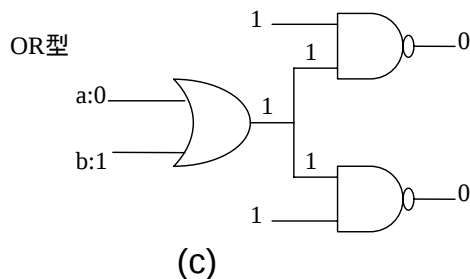


図 1. AND 型・OR 型ブリッジ故障

3. ドントケア抽出技術を用いたブリッジ故障検出の効率化

3.1 全体のフロー

本手法の全体のフローを図 2 に示す. まず初めに縮退故障用のテストパターンに X 抽出を適用することにより, ドントケアを含むテストパターンを生成する. 次にドントケアを含むテストパターンに対してブリッジ故障検出の効率化を行うようにドントケア部の再割り当てを行うことにより, ブリッジ故障の検出を増加させるテストパターンを生成する.

3.2 ブリッジ故障検出の効率化

本手法では, ドントケアに値を再割り当てすることによって, 新たに未検出のブリッジ故障を検出するテストパターンを生成し, ブリッジ故障の効率化を行う. ブリッジ故障検出の効率化についてのアルゴリズムを図 3 に記す.

Step1: ブリッジ故障シミュレーションを行うことによって, 縮退故障用のテストパターンで検出できないブリッジ故障を抽出する.

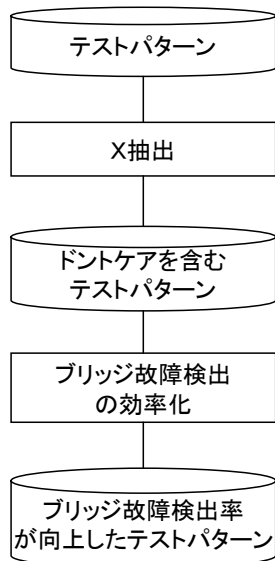


図 2. 全体のフロー

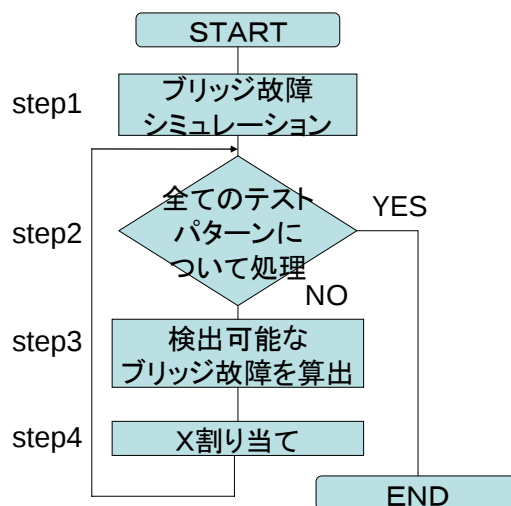


図 3. ブリッジ故障検出の効率化のアルゴリズム

Step2: 全てのテストパターンについてドントケアの再割り当てを行った場合処理を終了する. またドントケアに値を割り当てることによって, 検出可能なブリッジ故障がある場合には, Step3 へいく.

Step3: まだ未検出なブリッジ故障の中から, ドントケアに値を割り当てることにより検出可能になるブリッジ故障を解析する.

Step4: 検出可能なブリッジ故障に対してドントケアを割り当てる. ドントケア割り当てについての詳細は 4 章に記す.

3.3 ドントケアの再割り当て

本稿では, 縮退用のテストパターンに対してドントケアを割り当てることにより, 他の故障モデルであるブリッジ故障を検出する. 割り当ての方法は以下の三つのケースに分けられる.

Case1: 信号線の値の正当化のためドントケアに再割り当て

Case2: 縮退故障検出のためにドントケアに再割り当て

Case3: 信号線の値の正当化と, 縮退故障検出のためにドントケアに再割り当て

初めに Case1 の割り当てが可能かシミュレーションにより解析する. シミュレーションの結果 Case1 の割り当てが可能だった場合, Case1 の割り当て処理である, 信号線の値の正当化を行うようにドントケアに再割り当てを行う. Case1 の割り当てが不可能だった場合 Case2 の割り当てが可能かシミュレーションにより解析する. シミュ

レーションの結果 Case2 の割り当てが可能だった場合 Case2 の割り当て処理である、縮退故障検出のためにドントケアに再割り当てを行う。Case2 の割り当てが不可能だった場合 Case3 の割り当てが可能かシミュレーションにより解析する。シミュレーションの結果 Case3 の割り当てが可能だった場合 Case3 の割り当て処理である、信号線の値の正当化と、縮退故障検出のためにドントケアに再割り当てを行う。Case3 の割り当てが不可能だった場合、対象としているブリッジ故障は検出不可能である。

それぞれの Case における再割り当ての例を図 4 に記す。

図 4-a では信号線 d, e 間の OR 型ブリッジを検出するケースを考える。外部入力 (a, b, c, f, g) = (0, 1, x, 0, x) である。このテストパターンでは信号線 d の 1 縮退故障が検出可能であることが分かる。故に信号線 e に 1 を割り当てれば信号線 d, e 間の OR 型ブリッジ故障が検出可能であることが分かる。つまりこの場合 Case1 の割り当てを行うことができる。信号線 e のドントケアを 1 にするには外部入力である信号線 c の値を 0 に割り当てる。割り当て後のテストパターン (a, b, c, f, g) = (0, 1, 0, 0, x) は信号線 d, e 間の OR 型ブリッジ故障が検出可能となる。

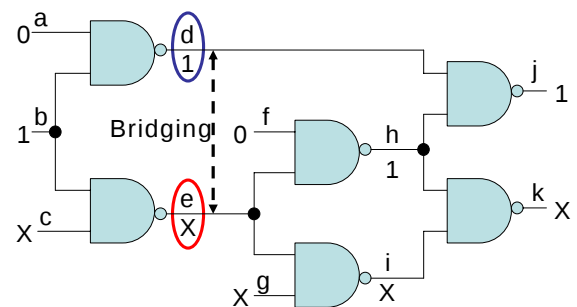
図 4-b は信号線 d, e 間の AND 型ブリッジを検出するケースを考える。外部入力 (a, b, c, f, g) = (0, 1, x, 0, x) である。このテストパターンでは信号線 d, e 共に 0 縮退を検出することができない、故に Case1 の割り当て処理は不可能である。しかし、信号線 d の値が 0 なので Case2 の割り当て処理が可能である。故に信号線 e に 0 縮退故障を検出するような値を割り当てれば、信号線 d, e 間の AND 型ブリッジ故障が検出可能であることが分かる。信号線 e の 0 縮退故障を検出するにはまず、信号線 e に 1 を割り当てるために外部入力 c のドントケアに対して 0 を割り当てる。さらに故障を外部出力まで伝搬させるため外部入力 g に 1 を割り当てる。割り当て後のテストパターン (a, b, c, f, g) = (1, 1, 0, 0, 1) は信号線 d, e 間の AND 型ブリッジ故障が検出可能となる。

図 4-c は信号線 d, e 間の AND 型ブリッジを検出するケースを考える。外部入力 (a, b, c, f, g) = (x, 1, x, 0, 0) である。このテストパターンでは Case1, Case2 両方の割り当て処理が不可能なので Case3 の割り当て処理を行う。まず初めに信号線 e の値を 0 にするように外部入力 c に 1 を割り当てる。さらに AND 型ブリッジ故障を検出するに

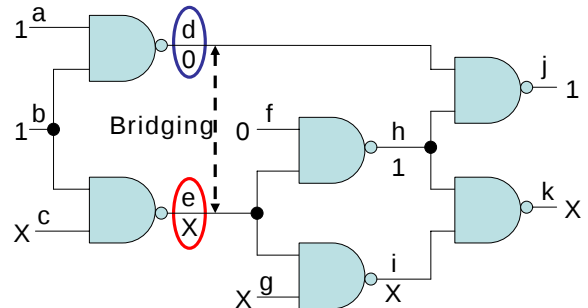
は信号線 d の 0 縮退故障を検出必要がある。信号線 d の 0 縮退故障を検出するために外部入力 a に 0 を割り当てる。割り当て後のテストパターン $\langle a, b, c, f, g \rangle = \langle 0, 1, 1, 0, 0 \rangle$ は信号線 d, e 間の AND 型ブリッジ故障が検出可能となる。

4. 実験結果

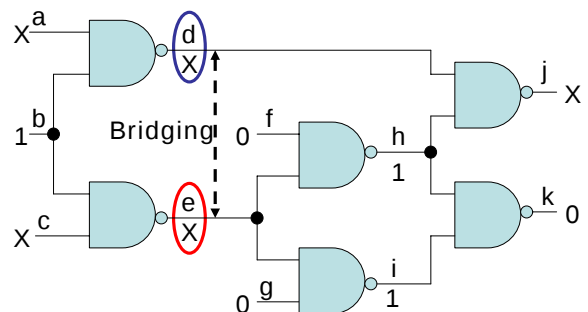
実験環境は OS が WindowsXP, CPU が Inter(R) Core(TM)2 CPU4300 1.8GHz, メモリーが 1.18 GHz の計算機を用いた。テストパターンは TetraMAX により得られた縮退故障用のテストパターンに対し、ドントケア抽出を行ったものを用いた。評価回路は ISCAS' 85 ベンチマーク回路を用いた。



(a) Case1



(b) Case2



(c) Case3

図 4. 値の再割り当て

表 1 は,ブリッジ故障シミュレーションの実験結果である.ブリッジ故障シミュレーションは縮退故障用のテストパターンに対して行った.実験結果を見てみると,縮退故障用のテストパターンでも,多数のブリッジ故障が見つかることがわかる.c2670 を除く全ての回路でブリッジ故障検出率が 90%以上であった.

表 2 は,割り当て前の縮退故障用のテストパターンのドントケアの割合である.c6288 の回路ではドントケアの割合が 1.7%と低い.一方 c5315 の回路では 63.5%と半分以上がドントケアで占めている.ドントケアの割合が高いほど割り当ての可能性が増えてくるので,ドントケアの割合が高いほどブリッジ故障検出の効率化が可能であることが分かる.

表 3 は,値の割り当て後のブリッジ故障検出率を示している.ドントケアの割合が少ない c499, C1355 では,ブリッジ故障検出数は上がっていないものの,他の全ての回路ではブリッジ故障検出数が上がっていることがわかった.特にドントケアの割合が高い c2670 においてはブリッジ故障検出率が 20%以上上がっているのがわかった.しかし割り当て数に比例して実行時間あがってしまう.

5. おわりに

本稿では,与えられた縮退故障用のテストパターンに対してドントケア抽出を行い,ブリッジ故障の U モデル検出を効率化するためのドントケアの再割り当て法を提案し,再割り当てしたテストパターンを評価した.

今後の課題として,ブリッジ故障の検出を効率化するための新たな割り当てを評価することが挙げられる.

参考文献

- 1)K. Miyase, S. Kajihara “XID: Don’t Care Identification of Test Patterns for Combinational Circuits,” IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, Vol. 23, No. 2, pp. 321-326,Fed. 2004
- 2)松雄三 塩坂知子 山田輝彦 山崎浩二“CMOS 回路における短絡故障の一モデルとそのテスト生成方”電子情報通信学会論文誌 D-I vol. J81-D-I No.6 pp.872-879 1998 年 6 月

3)Kohei Miyase¹, Kenta Terashima², Seiji Kajihara², Xiaoqing Wen² and Sudhakar M. Reddy“On Improving Defect Coverage of Stuck-at Fault Tests”

Proceedings of the 14th Asian Test Symposium(ATS ‘05) PP 216~223

表 1. ブリッジ故障シミュレーション実験結果

回路名	総ブリッジ故障数	ブリッジ故障検出数	ブリッジ故障検出率	実行時間 [sec.]
c432	9132	8509	93.18	0
c499	16681	16496	98.89	0
c880	81899	76067	92.88	0
c1355	90165	89353	99.10	0
c1908	307416	304347	99.00	1
c2670	1074617	838211	78.00	6
c3540	1241037	1204970	97.09	6
c5315	2977286	2890682	97.09	15
c7552	6696064	6641559	99.19	39

表 2. 各回路におけるドントケアの割合

回路名	ドントケアの割合[%]
c432	48.0
c499	3.4
c880	38.9
c1355	6.4
c1908	21.2
c2670	90.5
c3540	52.6
c5315	63.5
c7552	54.1

表 3. 値の再割り当て後のブリッジ故障検出率

回路名	総ブリッジ故障数	ブリッジ故障検出数	ブリッジ故障検出率	実行時間 [sec.]
c432	9132	8852	96.93	6
c499	16681	16496	98.89	0
c880	81899	81432	99.43	2
c1355	90165	89353	99.10	3
c1908	307416	307271	99.95	24
c2670	1074617	1074431	99.98	707
c3540	1241037	1240948	99.99	365
c5315	2977286	2977281	100.00	314
c7552	6696064	6696027	100.00	390