

クロストーク故障に対する検査入力生成手法について

板崎 徳禎, 徐 小楽, 樹下 行三
大阪大学工学部, 応用物理学科

あらまし

VLSIの高速化,高密度化に伴って、クロストーク故障と呼ばれる新しい論理故障が問題になってきている。本論文ではこのクロストーク故障に対する検査入力生成手法を提案する。クロストーク故障の検出には、連続する2つのテストベクトルが必要であり、一般的には検査入力生成は必ずしも容易でない。提案する手法では、2つのテストベクトルに対応する論理値を1つにまとめた信号値を用いて、簡明なアルゴリズムによって検査入力を生成することが可能となっている。また、ハザードなどの影響も考慮し、完全な検査入力生成を目指した。提案したアルゴリズムをC言語によってプログラム化し、いくつかの回路を用いて実験を行なった。

Test Pattern Generation for Crosstalk Faults

Noriyoshi Itazaki, Xiao Le Xu and Kozo Kinoshita
Department of Applied Physics, Faculty of Engineering, Osaka University
Yamadaoka 2-1, Suita, Osaka 565, Japan

Abstract

With the increasing of densities and switching speed of VLSI, a new type of logic fault called as cross talk faults becomes a problem. In this paper, we present a new method of test pattern generation to detect this cross talk fault. In the process of detection of the cross talk fault, generally successive two test vectors are needed. Therefore the test pattern generation is relatively difficult. However, since our algorithm uses special combined signal values corresponding two test vectors, test generation becomes simple. Moreover, by considering the affect of the hazard, complete test pattern generation can be expected. The algorithm was coded with C language and some experimental results were reported.

1. はじめに

近年、VLSIは益々高速で高密度なものとなってきており、これに伴って故障検査の面でも、従来の縮退故障モデルの範疇では取り扱えない、新しいタイプの故障が問題となっている。本論文では、それら新しいタイプの故障の中で、クロストーク故障 [1-6] と呼ばれる故障に注目し、その検査入力生成手法を提案する。

本論文で対象とするクロストーク故障は、正常な回路ならば本来影響を及ぼさないはずの信号線間で、製造上の問題などによる欠陥で電氣的な結合が密になり、回路内の信号線に起きる論理値の変化によって、他の信号線に論理値変化の影響が及んで誤った論理値が生成され、そのため回路が誤動作を起こす故障である。このようなクロストーク妨害が発生すると、非同期論理回路であればただちに回路内の他の部分に影響が及ぶことが考えられ、また同期式論理回路であっても、そのクロックラインやリセットラインなどに妨害が及んだ場合、回路が誤動作を起こす事が考えられる。

クロストーク故障の有無を検査するには、まずクロストークの源となる信号線（以後これを妨害信号線と呼ぶ）に0から1または1から0の論理値変化を与え、さらにその変化によるクロストーク妨害の有無を調べべき信号線（以後これを被妨害信号線と呼ぶ）から外部出力端子まで、妨害の有無を伝搬させる必要がある。そのため、一般的には連続する2つのテストベクトルを考え、最初のテストベクトルと次のテストベクトルとの境界で妨害信号線の論理値変化を与え、かつ2番目のテストベクトルはクロストーク故障の有無を外部出力端子に伝搬可能なように検査入力を定める必要がある。本論文で提案する検査入力生成アルゴリズムは、個々のテストベクトルをそのまま取り扱うのではなく、2つ1組のテストベクトルの論理値の変化点に着目して、1つの信号値で2つのテストベクトルの論理値とその変化点の情報を取り扱えるようにし、1回の検査入力生成によって、2つの連続するテストベクトルを生成するように考案されたものである。

また、クロストーク故障を考える場合は、クロストーク妨害を受ける信号線上にハザードがあると、クロ

ストーク妨害の現われ方が不安定になる恐れがあるので、検査入力生成時には被妨害信号線上にハザードが生成されないように、アルゴリズムを構成している。

対象回路として、ANDやOR素子からなるゲートレベルの組み合わせ論理回路を仮定し、クロストーク故障の有無を調べるべき1組の信号線は、レイアウト情報などから前もって与えられているものとしている。従って検査入力生成アルゴリズムは、任意の妨害信号線と被妨害信号線の組に対して、クロストーク故障の有無を検査できる入力を生成するものとする。この時、妨害信号線及び被妨害信号線は各々1本ずつと仮定する。また、回路内で同時に複数のクロストーク故障が存在することも仮定しない。

2. クロストーク故障

クロストークは、もともとアナログ通信路などで問題になっていたものであるが、近年VLSIの集積密度や動作速度が増加するにつれて、デジタル論理回路でも、クロストーク妨害による論理故障が問題になってきた。本来絶縁されているべき複数の信号線間でクロストークが現われるのは、容量性、誘導性、抵抗性などの結合が、信号線間などに存在することに起因すると考えられるが、これらの結合の程度が弱かったり、信号の時間変化率が小さい場合には、通常、論理回路の動作に影響を与えるようなクロストーク妨害は発生しにくい。しかし、製造時のなんらかの理由で線間容量が大きくなったり、GaAsなど高速の論理素子の場合などでは、クロストーク妨害による論理故障が発生しやすくなると考えられる [7,8]。クロストーク故障そのものの詳細については、上記各文献にゆずり、本論文では次に述べる仮定の上で検査入力生成の問題を扱うことにする。

図1 (a) に示すように、妨害信号線に論理値0から1に変化する信号値、または論理値1から0に変化する信号値が加わり、かつ被妨害信号線の論理値が、前者の場合は0、後者の場合は1の時、もしクロストーク故障が存在すれば被妨害信号線の論理値が0-1-0または1-0-1と変化するものと仮定する。しかしながら、図1 (b) に示されるように、妨害信号

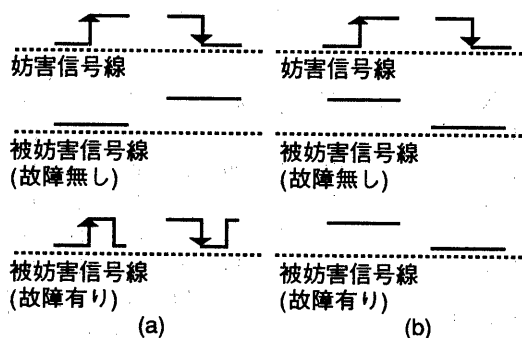


図1 クロストーク故障のモデル化

線に0から1（1から0）の変化があっても、被妨害信号線が1（0）の場合には、クロストーク故障の有無にかかわらず被妨害信号線の論理値は1（0）のままであると仮定する。被妨害信号線の信号値は、2つのテストベクトル間で変化のない論理値ばかりでなく、図2のように変化することも考え得るが、この場合、タイミングの漂動により、妨害信号線の論理値が変化する瞬間の被妨害信号線の論理値が、0なのか1なのか確定できなくなり、クロストーク故障の現れ方が不定になると思われるので、これを避けるため被妨害信号線の信号値は論理値0または1に固定する必要があるものとする。また同様の観点から、妨害信号線上では信号値変化の近傍でハザードを伴っていてもかまわないが、被妨害信号線上では妨害信号線の信号値変化の近傍で、ハザードが現われないことが必要であると仮定する。

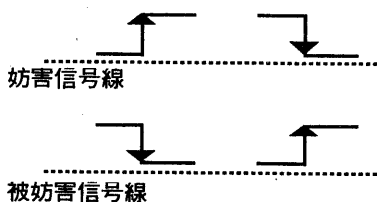


図2 クロストーク妨害が不定となる場合

クロストーク故障の存在による、被妨害信号線上での妨害パルスの時間中は、単一ハザードの中や信号値

変化の近傍でのハザード列の時間中と比較して十分広く、この妨害パルスを外部出力端子に伝搬して観測すれば、クロストーク故障の有無を容易に検出できるものと仮定する。

3. クロストーク故障の検査入力生成

クロストーク故障に対する検査入力を生成する場合、妨害信号線に論理値0から1、あるいは1から0の変化を与え、その変化の直後にクロストーク妨害の有無を外部出力端子まで伝搬させなければならないため、1つの故障を検査するために連続する2つの検査入力を求めなければならない。このため、スタックオープン故障を検査するための検査入力生成の場合と同じように、アルゴリズムが複雑になることが予想される。これを避けるため、検査入力の1パターン目の論理値、2パターン目の論理値および1パターン目から2パターン目の変化の情報を1つの信号値にまとめて取り扱い、アルゴリズムを簡明なものとした。また、論理値の変化に着目した信号値を用いるため、ハザードの有無が取り扱い易くなり、仮定したクロストーク故障のモデルの範囲内で、完全な検査入力生成を行なうことが可能となる。

3. 1 検査入力生成で用いる信号値

このアルゴリズムで使用する信号値を、表1に示す。0および1はそれぞれ論理値0と1をあらわす。SUは1パターン目の論理値が0で、2パターン目の論理値が1である信号値である。SDはSUの逆で、1パターン目が1、2パターン目が0を表す。P0は、1パターン目の論理値が1で、2パターン目の開始とともにクロストーク故障の影響で一定時間論理値が0になり、その後また論理値1にもどる信号値である。P1はP0の逆であり、0-1-0と変化する信号値である。DUは1パターン目は論理値0であり、2パターン目の最初でクロストーク故障の影響により、論理値0から1への変化が遅れている信号値を表す。DDも同様に、2パターン目の最初で論理値1から0への変化が遅れているものを表す。以上8つの信号値につ

表1 信号値表

記号	1パターン目の論理値	2パターン目の論理値	ハザードを伴う可能性
0	0	0	なし
1	1	1	なし
SU	0	1	なし
SD	1	0	なし
P0	1	0-1	なし
P1	0	1-0	なし
DU	0	0-1	なし
DD	1	1-0	なし
0H	0	0	あり
1H	1	1	あり
SUH	0	1	あり
SDH	1	0	あり
POH	1	0-1	あり
P1H	0	1-0	あり
DUH	0	0-1	あり
DDH	1	1-0	あり
U	未確定		

いては、1パターン目から2パターン目の変化点で、ハザードを伴う可能性のあるものを別に考え、記号の後ろにHを付けて表す。クロストーク故障の影響は、1パターン目から2パターン目の変化点より後に現われると仮定しているが、ハザードは変化点の両側に現われると仮定する。以上16の信号値が確定信号値である。最後の信号値Uは、未確定信号値を表し、他の外部入力値の変化で値が変わり得る信号値を表す。

3. 2 信号値の演算

前節で示した信号値について、ANDゲートの信号値演算を表2に示す。ORゲートなどの信号値演算についても、ANDゲートの場合と同様に容易に定めることができる。なお表2では、行と列が対称なので片側半分のみを示している。各ゲートの出力値は、そのゲートの任意の2入力の信号値を、対応する表の行と列にあてはめ、その交点に示された信号値を読むことで求められる。また多入力の場合も、2入力の場合と同様に、演算表を逐次適用して出力値を求めることができる。

表2 信号値演算表 (AND)

AND	0	1	SU	SD	P0	P1	DU	DD	U
0	0	0	0	0	0	0	0	0	0
1		1	SU	SD	P0	P1	DU	DD	U
SU			SU	0H	DUH	P1	DU	P1	U
SD				SD	SD	0H	0	SD	U
P0					P0	0H	DU	SD	U
P1						P1	0	P1	U
DU							DU	0	U
DD								DD	U
0H									U
1H									U
SUH									U
SDH									U
POH									U
P1H									U
DUH									U
DDH									U
U									U

AND	0H	1H	SUH	SDH	POH	P1H	DUH	DDH
0	0	0	0	0	0	0	0	0
1	0H	1H	SUH	SDH	POH	P1H	DUH	DDH
SU	0H	SUH	SUH	0H	DUH	P1H	DUH	P1H
SD	0H	SDH	0H	SDH	SDH	0H	0H	SDH
P0	0H	POH	DUH	SDH	POH	0H	DUH	SDH
P1	0H	P1H	P1H	0H	0H	P1H	0H	P1H
DU	0	DU	DU	0	DU	0	DU	0
DD	0H	DDH	P1H	SDH	SDH	P1H	0H	DDH
0H	0H	0H	0H	0H	0H	0H	0H	0H
1H		1H	SUH	SDH	POH	P1H	DUH	DDH
SUH			SUH	0H	DUH	P1H	DUH	P1H
SDH				SDH	SDH	0H	0H	SDH
POH					POH	0H	DUH	SDH
P1H						P1H	0H	P1H
DUH							DUH	0H
DDH								DDH

ハザードの演算に関しては、ハザードの可能性の有無について演算し、ハザードが生成されたり伝搬する可能性があれば、ハザードを伴った信号値として扱う。例えば信号値0Hと0Hの論理積の場合、非常に時間中の狭いハザードが、2つ以上の入力でもまったく同時に重なる可能性はあまり高くないと考えられるが、

そのような可能性がまったくないとも言えないので、出力にもハザードが現われるとしている。1Hと1Hの論理和の場合も、同様に出力は1Hとする。

また、例えばSUと0Hの論理積の場合、1パターンから2パターンへの変化点より前の部分のハザードが失われるが、出力値はハザードを伴っているとして扱う。

3. 3 検査入力生成アルゴリズム

アルゴリズムはPODEMを基礎としているが、PODEMが(1)D設定、(2)D伝搬という2つのステップで構成されているのに対し、提案するアルゴリズムは、(1)被妨害信号線への論理値0または1の設定、(2)妨害信号線への論理値変化の設定、(3)クロストーク故障の有無の伝搬という3つのステップから構成されている。また、この3つのステップを1通り完了することで、2つ1組の検査入力を求めることができるように、用いる信号値を定めている。

アルゴリズムの概要を次に述べ、さらに図3にフローチャートを示す。個々の仮定故障に対して、アルゴリズムはラベル「IO-1」に始まり、ラベル「検査入力生成」、ラベル「検査不能」のいずれかで終わる。

妨害信号線と被妨害信号線、さらに信号値SUによるクロストーク故障なのか、SDによるクロストーク故障なのかは、前もって与えられているものとする。

<テストパターン生成手続き>

[IO-1]

被妨害信号線に、クロストーク故障の存在を示す信号値P0またはP1が設定されているか否か調べる。設定されていれば、ラベルIO-2に進む。

まだ信号値が設定されていなければ、初期目標線を被妨害信号線とし、与えられている妨害信号値と対応する0または1の信号値を初期目標値としてラベルDIに進む。もしP0またはP1以外の信号値が設定されていれば、ラベルBTに進む。

[IO-2]

妨害信号線に妨害信号値(SUまたはSDまたはそれ

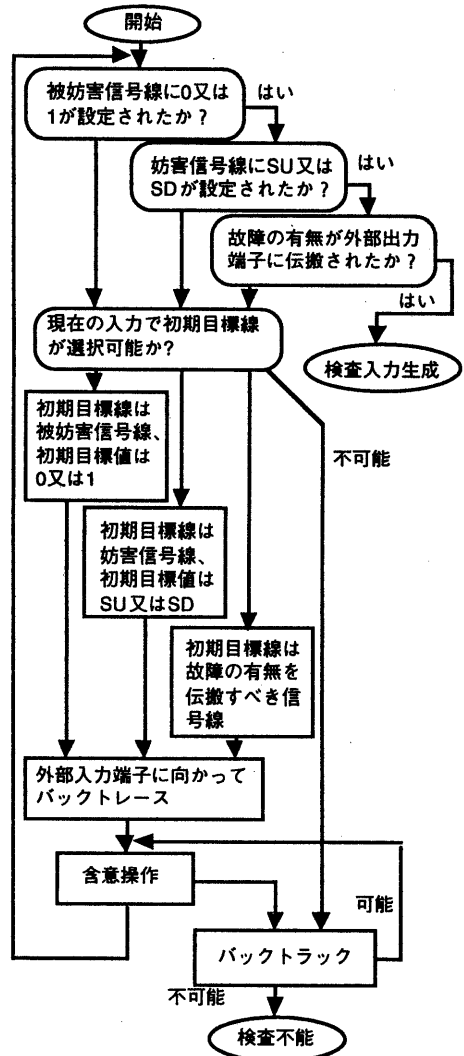


図3 フローチャート

らにハザードを伴った信号値)が設定されているか否か調べる。設定されていればラベルIO-3に進む。まだ信号値が設定されていなければ、初期目標線を妨害信号線とし、与えられている妨害信号値を初期目標値として、ラベルDIに進む。もし既に妨害信号値以外の信号値が設定されていれば、ラベルBTに進む。

[IO-3]

外部出力端子の内の任意の1つに、信号値P0, P1,

DUまたはDDが伝搬しているか否か調べる。伝搬していれば、ラベル検査入力生成に進む。伝搬していなければ、信号値P0, P1, DU, DDを次に伝搬させるべき信号線を初期目標線と定め、ラベルDIに進む。もしP0, P1, DU, DDを伝搬させ得る信号線がなければ、ラベルBTに進む。

[DI]

定まっている初期目標線、および必要に応じて初期目標値から、外部入力端子に向かって信号値Uとして残っている入力線の選択と目標値の割当をゲート毎に反復し、外部入力端子に信号値を割り当てる。この時割り当てられる信号値は、0, 1, SUまたはSDのいずれかである。その後ラベルDSに進む。

[DS]

新たに割り当てられた外部入力値と現在の回路内の信号値によって、新たに定まる信号値をすべて決定する。この過程で次の状態が発生すれば、以後どのように外部入力値の追加割当を行なっても検査入力は生成できないので、バックトラックを行なうため信号値決定を中断してラベルBTに進む。

- a) SU (SD) を設定すべき妨害信号線に、SD (SU) が設定された。
- b) 0 (1) を設定すべき被妨害信号線に、1 (0) が設定された。
- c) 被妨害信号線にSUまたはSDが設定された。
- d) 被妨害信号線にハザードを伴う信号値が設定された。

信号値決定が完了すれば、ラベルIO-1に進む。

[BT]

最も新しく割り当てられた外部入力値について、0, 1, SU, SDの中で、まだ試されていない信号値を割り当て、ラベルDSに進む。もし未試行の信号値がなければ、その外部入力の割当を取り消し、1つ前に割り当てられた外部入力について、未試行の入力値を割り当てる。すべての外部入力について、未試行の信号値が無くなれば、ラベル検査不能に進む。

[検査入力生成]

信号値P0, P1, DUまたはDDが伝搬している外部出力端子を観測端子とし、1パターン目の検査入力として、信号値0またはSUが割り当てられている外部入力端子では論理値0を、また信号値1またはSDが割り当てられている入力端子では論理値1を定める。2パターン目の入力としても同様に、1またはSUの場合には論理値1を、0またはSDの場合には論理値0を定める。

[検査不能]

仮定されたクロストーク故障に対しては、検査入力が存在しないとして検査入力生成を終了する。

4. 実験結果

クロストーク故障に対する検査入力生成アルゴリズムを、C言語によりプログラム化し、ゲートレベルのベンチマーク回路を用いて検査入力生成の実験を行なった。使用した計算機は、富士通S-4/LC型ワークステーションである。

本来、クロストーク故障の妨害信号線、被妨害信号線および妨害信号値の情報は、レイアウト情報などから与えられることを仮定しているが、現在手元にそのような回路情報がないので、ISCAS85のベンチマーク回路[9]を用い、上記妨害信号線と被妨害信号線の組を100組ランダムに選び、その各々について信号値SUによる妨害とSDによる妨害を仮定して、各回路ごとに200組ずつの仮定故障に対して検査入力の生成実験を行なった。なお、アルゴリズムそのものは、可能性のある入力値の組合せをすべて試し尽くすように構成されているが、実用上の観点から実際のプログラムでは、バックトラック回数による計算の打ち切りを行なっている。バックトラック上限値を200回にした場合の実行結果を表3に示す。表の検出の欄には検査入力が求めた故障の数が、冗長の欄には検査入力が存在しないと分かった故障の数が、中断の欄にはバックトラック回数の上限値を超えたので、テスト生成を打ち切った故障の数が示されている。また、CPUの欄には200組の仮定故障に対して要した生成

表3 実行結果

回路名	検出	中断	冗長	検出率	CPU[S]
C432	81	94	25	46.3	31.0
C499	128	67	5	65.6	25.4
C880	180	16	4	91.8	5.6
C1355	74	120	6	38.1	74.0
C1908	143	55	2	72.2	48.4
C2670	175	20	5	89.7	16.9
C3540	102	88	10	53.7	102.3
C5315	188	9	3	95.4	14.1
C6288	144	52	4	73.5	149.8
C7552	175	20	5	89.7	24.4

時間が秒の単位で示されている。

仮定したクロストーク故障自体が、ランダムに選ばれた信号線に対するものなので、実験結果について評価することはさほど有意義でないが、ここに示されたデータでは、90%程度以上の検出率を十分実用的な計算時間で得たものがあり、有用であると考えられる。半面、検出率が上がりにくい傾向の回路もいくつかあるが、これはPODEMでは外部入力値の割当を誤った場合、バックトラックによって0と1の2値の組み合わせを試し尽くせば良いのに対し、本アルゴリズムでは4値の組み合わせを試し尽くさねばならないので、より正確な外部入力値の割当が重要であることを示している。

5. あとがき

クロストーク故障を対象とした検査入力生成手法を提案し、さらにC言語によってプログラム化して、いくつかの回路で検査入力の生成実験を行なった。

今回述べたアルゴリズムでは、初期目標線から外部入力端子に向かって信号値の割当を行なう際やバックトラックによる入力値変更の順序などで、いくつかの自由度が残されており、ここに効果的なヒューリスティクスを導入すれば計算時間の短縮や検出率の向上を行なうことも可能と思われる。今後は実際のクロストーク故障の特徴なども考慮しながら、なるべく生成効率を増加させるように改良を行ないたい。

また、本稿ではクロストーク故障によって生じされ

たパルスの幅は、ゲートの伝搬遅延時間などと較べて十分広いと仮定しているが、デバイスの種類やクロストーク故障の発生形態によっては、ゲートの伝搬遅延時間などが無視できない場合も考えられる。今後は、このような時間因子を含めて対応を行ないたい。

参考文献

- [1] DEFALCO, J.A.: "Reflection and crosstalk in logic circuit interconnections," IEEE Spectrum, July 1970.
- [2] DEFALCO, J.A.: "Predicting crosstalk in digital systems," Computer Design, June 1973.
- [3] GOEL, A.K. and HUANG, Y.R.: "Modeling of crosstalk among the Ga-As VLSI connections," IEE Proc., Part G, 1989, vol.136, pp.361-368.
- [4] KONISHI, Y et al.: "Coupling noise between adjacent bit lines in megabits DRAMs," IEEE J on Solid State, Feb. 1989, pp.35-42.
- [5] MATTHAEI, G.L. and GONG, S.I.: "Simplified linear representation of logic gate terminal impedances for use in interconnect crosstalk calculations," IEEE J on Solid State, Oct. 1989, pp.1468-1470.
- [6] RUBIO, A., SAINZ, J.A. and KINOSHITA, K.: "Test pattern generation for logic crosstalk faults in VLSI circuits," IEE Proc., Part G, 1991, vol.138, No.2, pp.179-181.
- [7] ANGLADA, R. and RUBIO, A.: "An approach to crosstalk effect analysis and avoidance techniques in digital CMOS VLSI circuits," Int'l J of Electronics, 1988, vol. 65, pp.9-17.
- [8] ANGLADA, R. and RUBIO, A.: "A digital differential-line receiver for CMOS VLSI circuits," IEEE Trans. on Circuits and Systems, 1990.
- [9] BRGLEZ, F. and FUJIWARA, H.: "A Neutral redist of 10 combinational benchmark circuits and a target translator in FORTRAN," Special session on ATPG and fault simulator, ISCAS '85, June 1985.