

同期式順序回路のクロストーク故障に対する故障シミュレーション

井戸本康孝 板崎徳禎 樹下行三

大阪大学大学院 工学研究科 応用物理学専攻

〒565 大阪府吹田市山田丘2-1

TEL: 06 (879) 7835 FAX: 06 (879) 7836

E-mail: idomoto@ap.eng.osaka-u.ac.jp

あらし 同期式順序回路で問題となるクロストーク故障の検査を行うため、その故障モデルを考察し、さらに故障シミュレータを作成してランダム系列を用いた故障シミュレーション実験を行なった。本研究で対象とする故障は、回路内の許容量以上の容量結合の影響でクロック信号線にクロストークパルスが励起され、FFが誤動作を起こす故障である。クロストーク故障の検査では、論理値だけでなくタイミング動作を考慮する必要があるため、ユニット遅延モデルを用いて故障シミュレーションを行っている。ベンチマーク回路に対する故障シミュレーションの結果、故障の顕在化率及び検出率は、回路によって差が大きいが、ランダム系列を用いた場合で最高80%程度の検出率が得られる場合があることがわかった。

キーワード クロストーク故障, 故障シミュレーション, 同期式順序回路, クロック, ユニット遅延モデル

A Fault Simulation method for crosstalk faults in synchronous sequential circuits

Yasutaka Idomoto Noriyoshi Itazaki Kozo kinoshita

Dept. of Applied Physics, Faculty of Engineering, Osaka University

2-1 Yamadaoka, Suita, Osaka 565 Japan

TEL: +81-6-879-7835 FAX: +81-6-879-7836

E-mail: idomoto@ap.eng.osaka-u.ac.jp

Abstract In order to detect crosstalk faults that will cause strange errors for future high density synchronous sequential circuits, we have considered a fault model for the fault, and contrived a fault simulator based on the model. In this paper, we considered the crosstalk fault as a erroneous operations caused by crosstalk pulses originated by unexpected strong capacitive coupling between one data line and clock lines. Since we have to consider a timing in addition to a logic value, a unit delay model is introduced to our fault simulation. From our experiments on benchmark circuits, we can see that fault activation rates and fault detection rates are widely varied corresponding to circuit characteristics. Up to 80% fault detection rates are obtained from our simulation with random input vectors.

key words Fault simulation, Crosstalk fault, Synchronous sequential circuit, Clock, Unit delay model

1. 序論

近年の集積回路の微細化や高速化に伴い、従来の故障モデルの範疇では分類することのできない新しいタイプの故障が問題となってきた。本研究では、そのような新しい故障のうち、クロストーク故障と呼ばれる故障を取り扱う。

絶縁されている複数の導線間で、ある1本の、もしくは複数本の導線の電気信号の変化が、電磁氣的相互作用により、他の導線の電気信号に悪影響を及ぼす場合がある。この現象をクロストーク妨害といい、アナログ通信の分野においては、従来から問題となっていたが、論理回路中のクロストーク妨害についての研究はまだ少ない[1]-[4]。これは、論理回路では、0及び1のみのデジタル信号を扱うため、クロストーク妨害による信号線へのノイズの侵入が、従来はあまり問題にならなかったためと考えられる。しかし近年では、回路の設計ルールの微細化により、信号線間の間隔は狭まりつつある。また、信号線自体の幅もより狭くなりつつあるため、信号線断面の縦/横比が大きくなる傾向にある。このため、信号線間の結合容量は高まる傾向にある。また、高速スイッチング素子が導入されるに伴い、クロストーク妨害の影響が従来よりも現れやすくなってきている。これらの潜在的要因に加え、回路内の比較的長い距離を複数の導線が併走しているような場合には、それらの導線間の容量結合はさらに大きくなる。また、製造時の欠陥により、期せずして容量結合が増大する場合も考えられる。これらの要因から、論理回路においてもクロストーク妨害が回路動作に無視できない影響を及ぼす場合が考えられる。クロストーク妨害の強さが許容範囲を越え、その影響が過渡的な信号値異常として現れた場合、その回路は異常な動作を起こすことがある。このように、過度のクロストーク妨害のため、回路に信号値異常が発生するような故障を、クロストーク故障と呼ぶ。

クロストーク妨害は、それが発生する箇所によって、論理回路の動作に与える影響はさまざまである。クロストーク妨害によって誤った信号値が発生したとしても、正常回路と全く変わらない回路動作を示す場合もあれば、誤作動を起こす場合もある。リセット可能な同期式順序回路を考えた場合、回路は外部入力及び外部出力を含めたデータ信号線、FF (Flip-Flop) 内の信号線、VDD、GND、FFのリセット信号線、クロック信号線等から構成される。このうち、VDDやGNDは一般にインピーダンスが低いいため、クロストーク妨害による電圧変化は微少であり、回路動作には影響を与えないと考えられる。データ信号線に、他の信号線の信号値変化によって誤ったパル

スが励起された場合、このパルスは信号線を伝搬し、他の信号線や外部出力、FF入力に到達する可能性がある。このため、非同期式順序回路では、このパルスのために他の回路部分が誤動作を起こすことが考えられる。しかし、クロストーク妨害によって励起されるパルス信号（以後クロストークパルスと呼ぶ）の時間幅は、SPICEシミュレーションの結果などから、ゲート遅延時間程度であると考えられるため[2]、一般的な同期式順序回路を対象とする場合、クロック周期と最大回路遅延との差がごく僅かでない限り、データ信号線のクロストークパルスの問題は深刻ではないと考えられる。

同期式順序回路においてクロストークパルスが深刻な問題となるのは、FFのクロック信号線やリセット信号線等にクロストークパルスが励起される場合である。クロック信号線にクロストークパルスが励起された場合、1つまたは複数のFFの信号値が、通常動作中に誤って更新される場合が考えられる。また、リセット信号線にクロストークパルスが励起された場合には、1つまたは複数のFFが、通常動作中に誤って初期化される場合が起こり得る。しかし、クロック系統が通常動作時の高速スイッチング動作に対応する必要があるのに対し、回路の初期化時にのみ動作させるリセット系統には、必ずしも高速動作能力が必要であるとは限らないと考えられる。このため、リセット信号線とGNDの間にデカップリングコンデンサを付加したり、FFのリセット入力ゲートの入力インピーダンスを下げる等の手法により、リセット系統の低速化や低インピーダンス化と引き替えにノイズ性を高め、クロストーク妨害の影響を軽減できると考えられる。同期式順序回路における影響が深刻であり、なおかつ有力な対応策に乏しいと考えられるのは、クロックライン上に現れるクロストーク故障である。

そこで本研究では、クロック信号線に誤ったパルスが励起されるクロストーク故障を研究の対象とし、この故障に対する故障シミュレータをユニット遅延モデルに基づいて作成し、ランダムベクトルが印加された場合に、どれだけの故障が顕在化し、外部出力において検出されるかを調べた。

2. クロストーク故障のモデル

信号線間に起こるクロストーク妨害の原因は、容量性結合、誘導性結合、抵抗性結合の3種類が考えられるが、現在多用されているCMOS論理回路の動作に重大な影響を与えるクロストーク妨害は、主に容量性結合による導線間相互作用が原因であると考えられている[2]。そこで本研究では、導線間の許容量以

上の結合容量のために発生するクロストーク故障を対象とする。

クロストーク妨害による信号値異常に関与する信号線は、クロストーク妨害の原因となる信号値変化を起こす信号線と、その影響を受け、パルスノイズが励起される信号線の2種類に分けられる[4]。本研究では、前者を原因信号線、後者を被害信号線と呼ぶ。

クロストーク故障は、原因信号線と被害信号線の本数とそれらの組合せにより区別されるため、回路内に存在する故障が唯一組と仮定する単一故障に限った場合でも、対象となる故障は多数存在する。このため本研究では、原因信号線が回路のデータ信号線、被害信号線がクロック信号線であり、原因信号線と被害信号線がどちらも1本ずつであるような1:1クロストーク故障の、単一故障を対象とする。被害信号線に励起されるクロストークパルスによって誤作動を起こすFF（以後、被害FFと呼ぶ）の個数や組合せは、クロック系の実装方式により多様に変化すると考えられるが、本研究では1つのFFが被害を被ると仮定した場合と、全FFが被害を被ると仮定した場合についてそれぞれシミュレーションを行う。また、ファンアウトの幹を原因信号線とするクロストーク故障と、その枝を原因信号線とするクロストーク故障は、被害信号線が同一であれば等価であるため、代表故障としてファンアウトの幹を原因信号線とするクロストーク故障のみを対象とする。組合せ回路部分のゲート数を g 、FF数を f とすると、代表故障の総数は、被害FFを1つと仮定した場合は $g \times f$ 個、被害FFを全FFと仮定した場合は g 個となる。

FFを駆動するためには、クロック信号線に印加されるパルス信号は、ある程度の時間幅を持つ必要がある。例えば、 $1.25\mu\text{m}$ テクノロジーを用いた2層メタライゼーションCMOS回路では、 $8\mu\text{m}$ 幅の信号線間の線間容量は 10pF/cm から 20pF/cm 程度と計算され、並走距離が 1mm から 2mm である場合のSPICEシミュレーションの結果では、クロストークパルスの時間幅は 3ns 程度となり[2]、これはFFを駆動する能力があるものと考えられる。

クロストーク故障の被害信号線に励起されるクロストークパルスは、2種類に分けられる[2]（図1）。原因信号線の論理値が $0 \rightarrow 1$ に反転し、かつその時点で被害信号線が論理値0を保持している場合、被害信号線に正極性パルス（正クロストークパルス）が励起される。反対に、原因信号線の論理値が $1 \rightarrow 0$ に反転し、かつその時点で被害信号線が論理値1を保持している場合、被害信号線に負極性パルス（負クロストークパルス）が励起される。

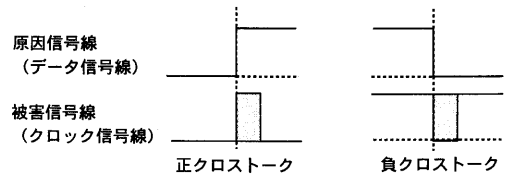


図1 クロストークパルス

3. 故障回路の動作

3.1 故障の顕在化と検出

順序回路にクロック及び入力ベクトルが印加されると、回路のデータ信号線は過渡状態を経た後、安定状態へと移行する。回路が過渡状態にある期間では、ベクトル印加による論理値反転やハザードにより、データ信号線に立ち上がり、立ち下りのエッジが現れる。故障回路の原因信号線にエッジが現れた場合、これと同期してクロストークパルスが励起され、本来クロックと同期して信号値を取り込むべき被害FFが誤作動を起こす。この時点の被害FFの入力線の信号値がその出力値と異なる場合、被害FFの出力が反転するため、ここに故障の影響が信号値の異常として現れる。このように、故障の影響により、ある時点においてFFの出力線に正常時とは異なる論理値が現れることを、本研究では故障の顕在化と呼ぶ。同一の被害FFにおいて、1タイムフレーム（クロック印加から次のクロックが印加されるまでの期間）内にFFの出力値の反転が偶数回起こった場合、FF値は初期値に戻るため、最終的な論理値のみを考慮すれば正常回路と等価な回路動作を示す。このような現象を、故障の自己隠蔽と呼ぶ。

故障の顕在化によって被害FFの出力線に現れた信号値異常は、組合せ回路部を伝搬し、入力ベクトル及び内部状態によっては、信号値異常が外部出力やFFに達する場合も考えられる。クロック周期が回路遅延と比較して十分に長いと仮定した場合、回路が完全に安定化した後に外部出力値が観測され、クロックが印加される。回路が安定化した状態で、故障による信号値異常が外部出力で観測される場合、その回路に故障が存在すると見なされる。外部出力の信号値異常は、必ずしも故障が顕在化したタイムフレームで観測されるとは限らず、回路の内部状態異常としていくつかのタイムフレームの間保持された後に観測される場合もある。

3.2 クロストークループ

故障の顕在化によって誤った値に反転した被害FFの出力値が、運悪くクロストーク故障の原因信号線へ伝搬した場合、その論理値反転によって再びクロ

ストークパルスが励起されることが考えられる。このように、データ信号線とFFが、クロストーク妨害を介して、一種のループ構造を形成する場合がある。回路内にこのようなループ構造（以下、クロストークループと呼ぶ）（図2）が形成されていると、FF値の反転と原因信号線の論理値反転が互いにそれぞれを励起し合い、信号値の振動が起こる可能性がある。本研究ではこの振動をクロストーク振動と呼ぶ。クロストーク振動は、ループの遅延とクロックのタイミングやデューティー比、入力系列等の性質により、すぐに消滅する場合もあれば、長期に渡り持続する場合もある。以下では被害FFがFFiの一つのみであると仮定した場合に、FFiと原因信号線 α の間のクロストーク妨害を介して、クロストークループが形成され、クロストーク振動が起こるための必要条件について、図2を用いて簡単にまとめる。

条件1 まず、はじめの故障顕在化によってQiに起こる論理値反転が、 α 及びDiへ伝搬しなければならないため、Qiから α へのパスとQiからDiへのパスは活性化されている。

条件2 Diが反転した後に少なくとも1回はFFiが誤動作を起こし、Diの値を取り込まなければならないため、 α に最後の論理値反転が起こるのは、Diの論理値が反転した後でなければならない。つまり、Qiから α への活性化パスの最大遅延は、最低限QiからDiへの活性化パスの最小遅延よりも大きい。

条件3 FFの誤動作を起こすクロストークパルスは正、負の2種類であり（図1）、クロストーク振動時には、これらが周期的に発生する。正、負クロストークパルスはそれぞれ、被害信号線が論理値1、0をとっている時のみ現れるため、クロストーク振動が持続するには、クロックがクロストークパルスを発生させるようなタイミングで印加される必要がある。

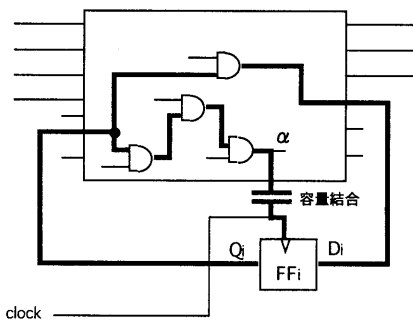


図2 クロストークループ

4. 故障シミュレーション

本研究が対象とするクロストーク故障は、その発生タイミングが重要であるため、伝搬遅延と、その影響によるハザードなど、信号波形情報を含めたシミュレーションを行う必要がある。表1には、今回の実験で用いたISCAS'89ベンチマーク回路[5]のデータを示している。入力側（外部入力及びFFの出力）から出力側（外部出力及びFFの入力）へ至る全パスのうち、最大のゲート段数を最大段数の欄に記載している。

4.1 波形の表現

信号線の波形の伝搬計算には、ビット演算を用いる。各信号線の波形は、図3のように0,1のビット列として計算機のワード内に格納し、右端のビットは外部入力にベクトルが印加される時点の論理値とする。今回使用したISCAS'89ベンチマーク回路では、最大ゲート段数の最大値はs641の74段であるから、クロストーク振動を考えなければ、74U（1Uは1ゲート分の単位遅延時間）の時間幅の波形を扱えば十分である。先に述べたとおり、クロストークループが形成されない限り、故障の顕在化はクロック印加後から回路の最大遅延時間以内で起こるが、クロストーク振動が発生すると、最大遅延時間以後に故障が励起されることがある。しかし、最大遅延時間以後に故障が励起された場合に必ず実動作時にクロストーク振動が発生するとは限らず、その発生は、実動作時のクロックの周期やデューティー比等に依存する。故障シミュレーションでこれらの判定を厳密に行うことは必ずしも容易ではないため、本研究では、回路の最大遅延時間の2倍の長さの信号波形を保持し、この期間での故障顕在化回数を、クロストーク振動の発生頻度の目安として記録する。例えばs641では、148段すなわち148Uの長さの波形を格納するため、各信号線の波形格納用に、32ビットのワードを5ワード割り当てる。

4.2 伝搬操作

波形伝搬の演算例を図4に示す、2入力ORゲートの入力に図のような波形が印加される場合、まずそれらのビット論理和演算を行う。各ゲートは遅延時間1Uを持つと仮定しているため、演算結果を1ビット左へシフトする。ビットシフト後の右端のビットには、入力ベクトル印加時とっていた論理値、つまり前タイムフレームの安定化後の論理値を入れる。以上の操作を行った後の波形がこのゲートの出力波形となる。ファンアウトの枝の波形は、ファンアウトの幹の波形をそのまま複写する。

波形の伝搬操作が必要となるのは、入力側からクロストーク故障の原因信号線へ至るパス上と、入力側

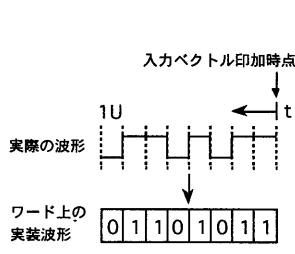


図3 信号波形の表現

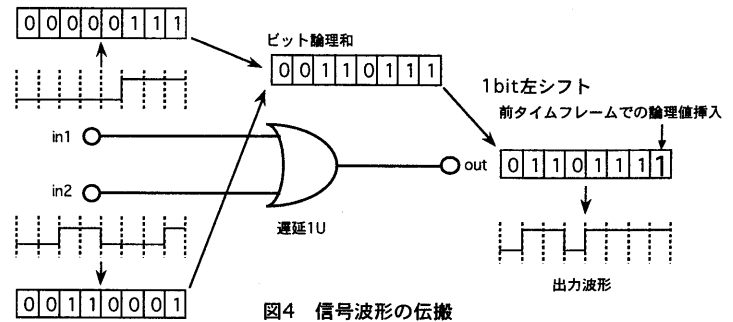


図4 信号波形の伝搬

から被害FFの入力へ至るパス上のみであり、その他の部分では波形の保持は不必要である。このため、波形を取り扱う必要のない回路部分では、通常の0遅延モデルの論理シミュレータと同様に、静的な論理値の伝搬操作のみを行い、実行時間の短縮を図る。

4.3 故障の顕在化判定と検出判定

伝搬操作が完了し、原因信号線の波形及び被害FFの入力波形が得られると、そのタイムフレームで故障が顕在化するかどうかを判定する。故障が顕在化する場合の例を図5に示す。原因信号線に立ち下がりエッジが発生し、被害信号線が論理値1を保持しているため、被害信号線には負クロストークパルスが励起される。被害信号線は短時間（クロストークパルス持続時間：1Uと仮定）の間論理値0を保った後、再び論理値1へ復帰する。この際に発生する立ち上がりエッジにより被害FFが誤作動し、その時被害FFの入力値1が取り込まれ、FF内伝搬遅延時間（1Uと仮定）後に出力される。取り込んだ値はそれまでのFF値0とは異なるため、FFの出力値が反転し、故障が顕在化する。

先に述べたとおり、故障が自己隠蔽を起こして自らの影響を落ち消すことがあるが、この場合でも、外部出力での出力波形を観測すれば、故障の影響が異常波形という形で検出される可能性がある。しかし本研究では、外部出力では安定状態の論理値のみを

観測するものとし、過渡状態時の波形の差異は故障検出には用いないとする。外部出力値の観測はクロック印加の直前とし、クロストーク振動が発生しない限り外部出力値観測時点までには回路は必ず安定状態になるものとする。

4.4 故障シミュレーションのアルゴリズム

アルゴリズムの説明の前に、これまで述べてきた仮定やパラメータ設定などについてまとめる。

- ・回路の伝搬遅延は、ユニット遅延を仮定する。
- ・使用するFFは正エッジトリガ型FFとする。
- ・FF内伝搬遅延時間（FFにクロックパルスが印加されてからFFの出力が変化するまでの時間）は1Uとする。
- ・クロストーク故障は、原因信号線がデータ信号線、被害信号線がクロック信号線であり、それらが各々1本ずつであるものを仮定する。
- ・1本の被害信号線上に励起されるクロストークパルスによって誤作動を起こすFFは、唯一つと仮定する場合と、回路内の全FFが誤作動を起こすと仮定する場合の双方を考える。
- ・クロック周期は、十分に長くと仮定する。このため、励起されるクロストークパルスは、負クロストークパルスのみとなる。
- ・回路の入力信号は、クロックと同期しており、FFの出力と外部入力値は、同時に組合せ回路部分の入力に入力されるものとする。
- ・クロストークパルスの持続時間は1Uと仮定する。
- ・原因信号線の過渡信号波形が、持続時間1U以下のパルスである場合は、FFの誤作動は起こらないと仮定する。

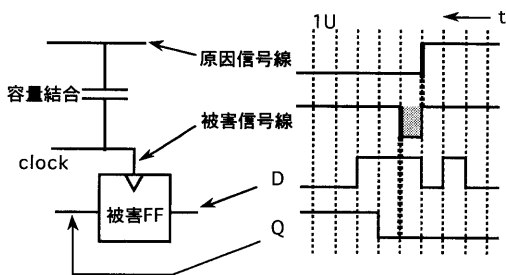


図5 故障の顕在化

本研究で提案する故障シミュレーションアルゴリズムは、与えられた対象回路と対象故障集合に基づき、各々の故障に対して、それが存在する回路に入力系列を印加する場合の回路動作及び出力応答をシミュレートし、故障が検出可能であるかどうかを判断するものである。出力としては、与えられた入力系列よ

る故障の顕在化率，検出率などを返す。また，一度顕在化された故障が，それ自身の影響を受けて隠蔽された回数や，クロストーク振動が発生した可能性などについてもレポートする。以下に，一つの故障に対する故障シミュレーションアルゴリズムの概略を，順を追って説明する。

1. ランダムベクトルを印加し，信号値の伝搬操作を行う。入力側から原因信号線へ至るパス上と，被害FFの入力へ至るパス上では，信号波形の伝搬操作を，その他の部分については，安定化後の論理値のみの伝搬操作を行う。
2. 故障が顕在化されるか否かの判定を行い，顕在化した場合は3.へ進む。顕在化しない場合で，以前に既に故障が顕在化されている場合は4.へ，その他の場合は5.へ進む。
3. 故障が顕在化した時点以降の被害FFの出力波形を書き直し，再び伝搬操作，故障顕在化の判定を行う。この操作は，故障の顕在化が起きなくなるまで繰り返す。
4. 安定化後の外部出力値を，正常回路の外部出力値と比較し，故障が検出されるか否かを判定する。故障が検出された場合は，その故障を検出可能としてシミュレーションを打ち切る。検出されない場合は，5.へ進む。
5. 印加した入力系列長が制限値を越えた場合は検出失敗とする。そうでなければ1.へ戻って新たなベクトルを印加する。

故障シミュレーションは，対象とする故障全てに対して行うが，実行時間の短縮のため，本研究では伝搬操作の一部をを並列的に行っている。回路中にどのような故障が存在する場合でも，故障が最初に顕在化されるまでの各信号線の信号波形は共通であるため，この期間の伝搬操作を一括して行い，順次顕在化判定を行う。大規模回路の1:1クロストーク故障に対しては，その故障数の多さから，全対象故障の顕在化や検出の情報を保持することが記憶容量的に困難であるため，同一のFFを被害FFとする故障を1つのグループとし，各グループ毎に並列伝搬操作を用いてシミュレーションを行う。

5. 故障シミュレーション実験

提案した故障シミュレーションアルゴリズムをC言語でプログラム化し，ISCAS'89ベンチマーク回路を用いてシミュレーションを行った。使用した計算機は，富士通S-4/5である。入力系列としては，今回はランダムベクトルを用いている。今回作成したプログラムは，40000本までの内部信号線を持つ回路中の全故障に対して，10000ベクトルまでの系列長の故障

シミュレーションを行うものであり，占有メモリは約6.3MBである。

クロストーク故障は，接近した信号線間の容量結合が原因であるため，離れた位置にレイアウトされている信号線間には故障は存在しない。提案したアルゴリズムでは，対象とする故障の集合が予め与えられており，それらに対して順次シミュレーションを行うことを仮定しているが，本来はレイアウト情報等を基に対象故障のある程度絞り込み，対象故障集合を作成すべきである。しかし，本研究では手元にそのような回路情報がないため，対象故障の選定の際にレイアウト情報は考慮せず，回路の静的構造解析によって判定される顕在化不能故障のみを対象から除外し，その他の故障は全て故障シミュレーションを行う対象とする。

5.1 実験1

表2は，1つの故障につき被害FFが唯1つと仮定した場合のシミュレーション結果である。この場合の総故障数は，(回路の総ゲート数 g) \times (FF数) である。s9234までの比較的小規模な回路については，顕在化不能であることが回路の静的構造解析から明らかな故障については対象から除外し，それ以外の故障を対象故障としてシミュレーションを行った。除外した顕在化不能な故障とは，入力側から原因信号線までの最大パス遅延が，入力側から被害FFの入力までの最小パス遅延よりも小さい故障である。これらの故障では，原因信号線の信号値が安定状態となる以前に，被害FFの入力値が変化し得ないため，故障が顕在化されることはなく，従って回路動作に悪影響を与えない。

一方，s15850では，1故障当たりのシミュレーション時間が長く，総故障数も多いため，明らかな顕在化不能故障を除いた対象故障集合から，各FF毎に，そのFFを被害FFとする故障を10個ずつランダムに選択し，対象故障とした。従って，対象故障数は (FF数) \times 10である。

総顕在化回数の欄にはシミュレーション実行中に故障が顕在化された回数，つまりFFの出力が誤作動によって反転した回数を示した。自己隠蔽回数の欄には故障の自己隠蔽が発生した回数を示し，顕在化故障数の欄には，少なくとも1回は顕在化された故障の数を示した。顕在化率と検出率は，それぞれ顕在化故障数と検出故障数の，対象故障数に対する割合である。振動の欄には，クロックが印加されてから，回路の最大遅延時間より遅い時刻に故障が励起された回数を示した。これは，1度励起された故障の影響が原因信号線に伝搬し，再び故障を励起したことを意味しており，4.1で述べたように，クロストーク

振動発生疑いの疑いがある。CPU平均の欄には、シミュレーション時間の、1故障あたりの平均値を示した。

5.2 実験2

表3は、被害FFを回路の全FFと仮定した場合、つまり故障の影響でクロック信号線にクロストークパルスが励起されると、回路の全FFが同時に誤作動すると仮定した場合のシミュレーション結果である。この場合の総故障数は、回路の総ゲート数と等しい。この実験では、実験1のような、明らかな顕在化不能故障の除去は行っていないため、対象故障数は総故障数に等しい。その他の項目については実験1に準じる。

5.3 実験3

入力系列の系列長と故障検出率との関係を、いくつ

表1 ベンチマーク回路のデータ

回路	外部入力数	外部出力数	FF数	ゲート数	最大段数
s344	9	11	15	160	20
s386	7	7	6	159	11
s444	3	6	21	181	11
s510	19	7	6	211	12
s526	3	6	21	193	9
s641	35	24	19	379	74
s820	18	19	5	289	10
s838	35	2	32	390	56
s953	16	23	29	395	16
s1196	14	14	18	529	24
s1423	17	5	74	657	59
s1488	8	19	6	653	17
s5378	35	49	179	2779	25
s9234	19	22	228	5597	58
s15850	14	87	597	9772	82
s35932	35	320	1728	16065	29

表2 実験1の結果 (被害FFを1個と仮定, 1000ベクトル)

回路	総故障数	対象故障数	総顕在化回数	自己隠蔽回数	顕在化故障数	顕在化率 (%)	検出故障数	検出率 (%)	振動	CPU平均 (ms)
s344	2400	2140	5938	8	1523	71.2	1419	66.3	2	41
s386	954	489	446	0	112	22.9	94	19.2	0	48
s444	3801	2735	12019	105	354	12.9	0	0	3	99
s510	1266	982	1121	4	552	56.2	552	56.2	2	15
s526	4053	2858	18339	0	334	11.7	0	0	0	115
s641	7201	5279	9290	1	2590	49.1	1911	36.2	0	218
s820	1445	1150	2045	0	630	54.8	593	51.6	0	67
s838	12480	9816	819	1	129	1.3	109	1.1	0	19
s953	11455	9657	8455	3	4222	43.7	4220	43.7	3	15
s1196	9522	8503	103121	1	7422	87.3	4204	49.4	0	873
s1423	48618	39164	409930	40	18514	47.3	4765	12.2	0	981
s1488	3918	3918	6446	104	2904	74.1	2778	70.9	4	102
s5378	497441	335794	1089662	5	120498	35.9	84226	25.1	0	789
s9234	1276116	797331	470722	36	15882	2.0	707	0.09	0	285
s15850	5833884	5970	5493	0	161	2.7	1	0.02	0	15139

表3 実験2の結果 (被害FFを全FFと仮定, 1000ベクトル)

回路	対象故障数	総顕在化回数	自己隠蔽回数	顕在化故障数	顕在化率 (%)	検出故障数	検出率 (%)	振動	CPU平均 (ms)
s344	160	311	0	140	87.5	140	87.5	1	12
s386	159	123	0	44	27.7	44	27.7	1	16
s444	181	5797	0	51	28.2	0	0	4	438
s510	211	336	0	165	78.2	165	78.2	1	15
s526	193	7575	0	53	27.5	0	0	0	552
s641	379	751	0	277	73.1	263	69.4	0	113
s820	389	419	1	169	58.5	164	56.7	0	72
s838	390	494	1	75	19.2	70	17.9	0	131
s953	395	717	0	355	89.9	355	89.9	4	17
s1196	529	2155	0	475	89.8	465	87.9	0	204
s1423	657	1092	0	493	75.0	486	74.0	0	110
s1488	653	1170	1	575	88.1	575	88.1	7	19
s5378	2779	3663	0	1806	65.0	1806	65.0	11	38
s9234	5597	32546	0	1340	23.9	871	15.6	0	3164
s15850	9772	155275	0	2091	21.4	797	8.2	8	16132
s35932	16065	23902	0	11925	74.2	11925	74.2	15	406

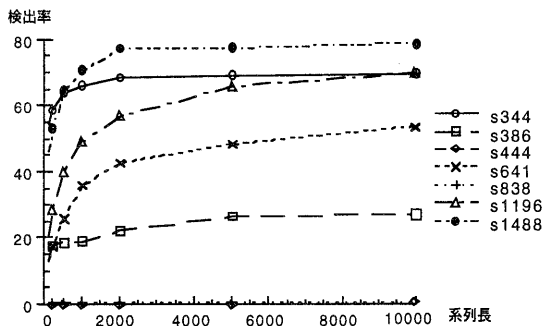


図6 系列長と故障検出率の関係 (実験3)

かの小規模回路について調べた。この結果を図6に示す。図から、多くの回路で10000ベクトル程度の系列長で検出率はほぼ飽和していると見られる。実験1, 2では、計算機的能力を考慮し、系列長の制限値を全回路一律に1000ベクトルとしたが、ランダムベクトルによる故障検出能力を見るには、10000ベクトル程度に渡る故障シミュレーションを行うことが望ましいと思われる。

5.4 実験結果の考察

実験結果の顕在化率を見ると、実験1で1.3%~87.3%、実験2で19.2%~89.9%と、どちらの場合も故障顕在化率は回路によって大きく異なる。故障検出率は、多くの場合故障顕在化率と同等か、多少低い程度の水準となっているが、回路によっては故障顕在化率よりはるかに低い場合や、故障が全く検出されない場合もある。この要因としては、故障が顕在化する被害FFが特定のものに限定されており、その被害FFが外部出力からみて深い位置にあり、故障の影響が観測されにくいことが考えられる。故障顕在化率及び故障検出率を引き上げるためには、故障を顕在化するための入力系列と、顕在化した故障の影響を外部出力へ伝搬させるための入力系列とを、アルゴリズム的に求める手法等の開発が必要であると思われる。

また、今回の実験で対象とした故障の中には、本質的に顕在化不能であるものが含まれていると考えられ、これらの故障が顕在化率、検出率を引き下げている可能性がある。従って、今回の実験での顕在化率及び検出率が低いからといって、ランダム入力系列がその回路の故障検査に向いていないとは、一概には断定できない。顕在化不能故障は、通常動作時において回路の誤動作を引き起こすことがないため、これらを対象故障内に含むことは無意味である。故障シミュレーションを無駄なく行うためには、レイ

アウト情報から対象故障を絞り込むと同時に、顕在化不能故障を指摘して対象故障を削減することが有効であると考えられる。実験1では、回路の構造解析によって、明らかに顕在化不能である故障の削除を行ったが、より詳しい回路の解析を行い、顕在化不能故障を指摘の手法の開発が必要であると思われる。

7. まとめ

クロストーク妨害が同期式順序回路に与える影響を考え、特に問題であると考えられる、クロックライン上のクロストーク故障を対象とした故障シミュレータを作成し、ランダム入力系列を用いた故障シミュレーション実験を行った。

クロストーク故障の動作を正確にシミュレートするためには、より正確な遅延モデルを導入することが望ましいが、本研究では、計算機的能力を考慮した上で、比較的簡単に扱うことのできるユニット遅延モデルを導入している。また、信号波形を0, 1のビット列で表現し、伝搬計算にはビット演算を用いた。各パラメータの定義や仮定については、さまざまな組合せが考えられるが、今回のシミュレーション実験では、それらの組合せの中の1つを用いている。

実験の結果、故障の顕在化率及び検出率は回路によって開きが大きく、極度に故障顕在化率が低い回路や、顕在化された故障が全く検出されない回路もあった。より効率よくクロストーク故障の検査を行うためには、アルゴリズム的なテスト生成手法等の開発が必要であると思われる。

参考文献

- [1] J. A. Defalco, "Reflection and crosstalk in logic circuit interconnections", IEEE Spectrum, 7, 7, pp. 44-50 (1970).
- [2] A. Rubio, N. Itazaki, X. Xu and K. Kinoshita, "An approach to the analysis and detection of crosstalk faults in digital VLSI circuits", IEEE Trans. CAD, 13, 3 pp. 387-395 (March 1994).
- [3] 板崎徳禎, 内藤久生, 樹下行三, "ゲート伝搬遅延を考慮したクロストーク故障の検査入力生成手法", 電子情報通信学会論文誌 DI, Vol. J77, No.5, pp.384-391 (May 1994).
- [4] R. Anglada and A. Rubio, "An approach to crosstalk effect analysis and avoidance techniques in digital CMOS VLSI circuits", Int'l J. of Electronics, 65, pp. 9-17 (1988).
- [5] F. Brglez, D. Bryan, K. Kozminski, "Combinational profiles of sequential benchmark circuits", Proc. of International Symposium on Circuits and Systems, pp. 1929-1934 (1989).