

パス遅延故障のテストにおけるパス選択手法について

村上 敦† 梶原 誠司† 笹尾 勤† スダーカ M. レディ†

†九州工業大学 情報工学部 電子情報工学科

‡アイオワ大学 電気コンピュータ学科

〒 820-8502 福岡県飯塚市大字川津 680-4

E-mail: murakami@aries30.cse.kyutech.ac.jp, {kajihara,sasao}@cse.kyutech.ac.jp,

reddy@eng.uiowa.edu

あらし

高性能な VLSI では、高い信頼性を保証するために遅延故障のテストが重要となる。しかし遅延故障のテストに有効なパス遅延故障モデルには、バス数が回路規模に対して指数関数的に増大する問題がある。パス遅延故障のテストを実現するためにはテストの対象となるバスを限定する必要があるが、バス数を十分に限定出来ない、またはテスト対象のバスにテスト不能なバスが含まれる等の問題がある。本論文では、より有効なパス選択手法として、各信号線に対して機能的活性化可能な最大遅延を与えるバスを抽出する手法を提案する。実験では、提案手法によってより小さなテストの対象となるバス集合を得られたことを示す。

キーワード 遅延故障のテスト, パス遅延故障, 機能的活性化不能バス, 部分経路活性化, テスト生成

Path Selection for Path Delay Testing

Atsushi MURAKAMI† Seiji KAJIHARA† Tsutomu SASAO† Sudhakar M.Reddy†

†Computer Science and Electronics Dept., Kyushu Institute of Technology

‡Electrical and Computer Eng. Dept., University of Iowa

680-4 Kawazu, Iizuka, Fukuoka 820-8502 Japan

E-mail: murakami@aries30.cse.kyutech.ac.jp, {kajihara,sasao}@cse.kyutech.ac.jp,

reddy@eng.uiowa.edu

Abstract

For high-performance VLSIs, delay fault testing has gained increasing the importance for circuit's reliability. However, the previous methods to select paths for path delay testing which can powerfully give guarantees on the behavior of the circuit has mainly two problems. One is that these methods cannot reduce paths enough for test generation, and the other is that a path set occasionally contains a large number of untestable paths. In this paper, we propose a more efficient method for path selection that based on the approach to select the longest functionally sensitizable paths for each line. Experimental results show that the proposed method can earn a smaller set of paths to be tested.

key words delay testing,path delay fault,functionally unsensitizable,partial path sensitization,test generation

1 はじめに

同期式回路では、製造された回路の遅延時間が定められたクロック間隔を越えた場合、回路に誤動作が生じる。このような回路の動作速度に依存する故障を遅延故障と言うが、遅延故障のテストは特に高性能な VLSI において重要な課題となってきた。

遅延故障の故障モデルとしてはバス遅延故障モデル [Smit85] とゲート遅延故障モデル [Barz83] が提案されている。ゲート遅延故障モデルは、ゲートのスイッチングの際の遅延時間の増大を仮定するモデルである。これに対してバス遅延故障モデルは、論理回路の外部入力 (またはフリップフロップ) から外部出力 (またはフリップフロップ) に至るバス上に蓄積する遅延をモデル化する。ゲートの遅延のみならず配線の遅延も考慮しているため、バス遅延故障モデルは実際の回路に生じるより多くの故障を検出することが可能であり、高品質なテストを実施できる。

バス遅延故障のテストでは、論理回路内の全てのバスをテストすることが望ましい。しかし、論理回路のバス数は回路規模に対して指数関数的に増大する場合があり、全てのバスについてテストすることは現実的には許容されない。故障シミュレーション的な手法で、与えられたテストパターンに対してバス遅延故障時の検出率の下限をバス数に依存しない計算量で求める手法は存在する [Pome92] が、未検出故障のテストを如何に実施するかという問題が残る。また、論理回路は信号の遷移を伝搬出来ないバスを多く含んでいることもわかってきた。そのため、テスト生成の対象となるバスを限定することが必要になってきた。これらの研究では主に二つのアプローチが取られている。

一つは、テスト対象となるバスを特定し、バス数を現実的にテスト実行可能な程度に限定する手法である。この手法では回路内の各信号線に対して、その信号線を含む最も遅延の大きなバスが含まれるようにバス集合を求める [Li89]。この考え方は、バス集合の要素数が最大でも回路の信号線数であるため、テストの対象となるバスの数は、総バス数と比較すると十分に小さい。しかし、大きな遅延を与えると考えられるような単純に長いバスの多くはテスト不能バスであることが多く、テストの対象として適切ではない。

もう一つのアプローチは、テスト不要なバスを前もって識別することによってテストすべきバスの集合を小さくする方法である [Spar95][Chen93][Lam93]。これら

の研究では論理回路には遅延故障のテストが不要なバスを多く含むものが存在することが指摘されている。それらのテスト不要なバスを識別し、テストの対象から除外することによって、より小さな対象故障の集合を得ることができる。テスト不要なバスのクラスとしては、機能的活性化不能 (Functionally Unsensitizable) バス [Chen93]、ロバスト依存 (Robust Dependent) バス [Lam93] などがある。バス数の大きな回路に対してこれらのテスト不要なバスを識別することは多大な時間を要し、例えば ISCAS'85 ベンチマーク回路の c6288 のように 10^{20} 本以上のバスが存在する回路では実用的な時間内に計算を完了することが出来なかった。これに対して、バス数に依存しない手法 (部分経路活性化手法 [Kaji97]) が提案され、その後もこの手法に基づいたテスト不要バスの識別法が開発されており [Li97][Hera97]、効率的にテスト不要なバスを識別することが可能となっている。しかし、依然としてテストすべきバスが多数残っており、FU バスを識別するだけではバス遅延故障のテストの実現は困難である。

本論文では、これら二つのアプローチを一つのフレームワークに組み込んだ、バス遅延故障のテストのためのバス選択アルゴリズムを提案する。本論文では、テスト可能な最長バスを抽出するアルゴリズムと、各信号線に対してその信号線を含む最も長いバスの集合を計算するためのフレームワークを開発する。

提案手法は部分経路活性化手法を用いてテスト不能なバスを識別する。そのため、テスト不能バスの識別に必要な時間が短時間で済む。また、テスト可能バスのみをテストの対象として選択するため、選択されたバスがテスト不要である可能性が低い。さらに、選択されたバスの集合は各信号線に対して最大遅延を与えるバスを少なくとも一本は含むため、多くの選択されなかったバスをカバーできるような、小さなバス集合を得ることが可能である。

以下、本論文は次のように構成される。2章では、本論文における用語等の諸定義、本論文の手法が用いている機能的活性化不能バス識別手法である部分経路活性化手法、および本論文における手法の概略について概説する。3章では本論文の手法におけるバス選択アルゴリズムを説明する。4章にベンチマーク回路に対する実験結果を示し、5章で本論文のまとめを行う。

2 準備

2.1 諸定義

物理パス $P = (l_0, l_1, \dots, l_m)$ は外部入力 l_0 から外部出力 l_m に至る信号線の系列である。分岐の枝は、分岐の幹の信号線と別の信号線とする。論理パス Px は、物理パス P の外部入力 l_0 において信号の遷移 $x \in \{\text{rise, fall}\}$ を持つようなパスを表す。信号線 $l_i (0 \leq i \leq m)$ は、 P のパス上入力 (on-path-input) と呼ばれる。パス上入力 l_i が入力しているゲートのファンイン信号線で、パス上入力でない信号線を、 Px のパス上入力 l_i に関するパス外入力 (off-path-input) という。ゲート入力の制御値とは、そのゲートに入力されている他の信号値に関わらず出力の値を決めることができるゲート入力の信号値である。例えば、信号値 0 は、AND および NAND ゲートの入力の制御値である。制御値でない信号値を非制御値と呼ぶ。部分物理パス $PP = (l_s, l_{s+1}, \dots, l_t)$ は、外部入力または分岐の枝 l_s から外部出力または分岐の幹 l_t に至る信号線の系列であり、パス上の信号線 $l_i (s < i \leq t)$ は分岐の枝を含まない。部分物理パスは分岐の枝 l_s が特定されることによって一意に定まる。部分論理パス PPx は、 l_s において信号の遷移 $x \in \{\text{rise, fall}\}$ を持つ部分物理パスである。以下では、部分パスとの区別を明確にするため、外部入力から外部出力までのパスを完全パスと呼ぶ。また、以下ではパスに関して、遅延時間の大きなパスを長いパスと表現する。最長パスは蓄積遅延時間が最大となるパスを指す。

2.2 提案手法の概略

パス遅延故障のテストでは、回路のパス数が非常に大きくなる場合がある。そのため、全てのパスをテストの対象とすることは現実的に許容されない。この問題に対する戦略の一つとして、回路の各内部信号線に対して最も大きな遅延を与えるパスをテストの対象とする考え方があ。しかし、単純に遅延の大きなパスは、機能的活性化不能である場合が多いため、テストの対象として適切ではない。よって本研究では、より有効な戦略として、各信号線に対して、機能的活性化可能なパスのうち最大遅延を与えるパスをテストの対象とする。

本研究における機能的活性化可能パスの識別には部分経路活性化手法を用いる。部分経路活性化手法は、機能的活性化不能パスを高速に発見するために開発された手

法である。部分経路活性化手法については 2.3 節で述べるが、b-f ペアと呼ばれる、二つの信号線を含む全ての完全パスが機能的活性化不能であるような分岐枝のペアを発見する。

本論文では、部分経路活性化手法によって得られる結果を基に、次の二つの手順を開発する。

1. ある信号線を通る機能的活性化可能パスで最大遅延を与えるものを抽出し遅延時間を計算するアルゴリズム
2. 全ての信号線に関して、機能的活性化可能な最大遅延を与えるパスが含まれるパス集合を求めるためのフレームワーク

アルゴリズム (1) では、パスの数え上げに基づいて、活性化できるパスとそうでないパスに分類し、最長パスとその蓄積遅延時間とを結果として出力する。

アルゴリズム (2) は、アルゴリズム (1) を適切な順序で回路内の各信号線に適用するために開発する。共通のパスを機能的活性化可能な最長パスとして持つことが分かっている複数の信号線については、そのいずれか一つの信号線に対してアルゴリズム (1) を適用すればよい。本研究ではこの観測に基づく効率的なフレームワークを提案する。

2.3 部分経路活性化手法 [Kaji97]

部分経路活性化手法はまず部分論理パスに回路を分割する。次に各部分論理パスについて、部分論理パスが機能的活性化可能であるための必要割当を計算し、それらを解析することによって、b-f ペアと呼ばれる、それらを含むパスが機能的活性化不能であるような信号線のペアを求める。ある論理パスがロバスト依存パス集合に含まれるための条件は、[Spar95] の Lemma 2 によって、パス上の各ゲートのパス外入力に対する条件として与えられる。部分経路活性化手法は、部分論理パスがそれらの条件を満たすための必要割当を計算し、次の補題に基づいてテスト不能パスを求める。

Lemma 1 [Kaji97]: 部分論理パス PPx と PQy に関して、信号線 l の論理値 $v \in \{0, 1\}$ が、 PPx が機能的活性化可能であるための必要割当であり、かつ、信号線 l の \bar{v} が PQy が機能的活性化可能であるための必要割当であるとき、 PPx と PQy とをともに含む全ての完全論理パスは、機能的活性化不能パスである。

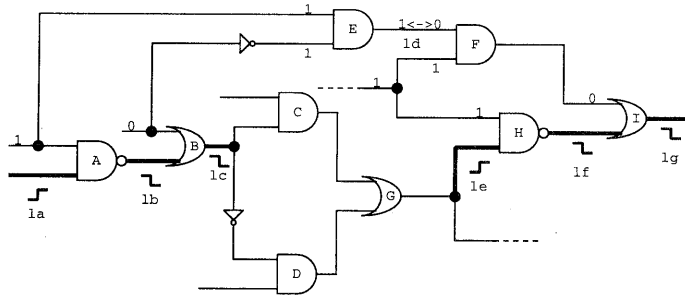


図 1: 部分経路活性化手法

Lemma 1 における PPx と PQy のそれぞれの始点となる信号線 (信号値の遷移に関する情報を含む) のうち、外部入力側の信号線は b-line と呼ばれ、残る一方の信号線は f-line と呼ばれる。これらの信号線のペアを b-f ペアと呼ぶ。

例を図 1 に示す。部分パス $PPx = (l_a, l_b, l_c)$ および $PQy = (l_e, l_f, l_g)$ について、 l_a, l_e にそれぞれ立上り信号を仮定する。このとき、信号が伝搬するための必要割当は l_d において矛盾が生じる。したがって、 PPx と PQy とを同時に活性化することはできない。したがってこの部分パスの始点の信号線 l_a が b-line、 l_e が f-line となり、 (l_a, l_e) で一つの b-f ペアを構成する。

部分経路活性化手法は各部分論理パスの必要割当を比較することによって b-f ペアを求める。b-f ペアを含むパスは機能的活性化不能である。部分経路活性化手法は、部分パスを活性化する必要割当を求め比較するため、時間計算量は高々 $O(N^2)$ であり、完全パス数 $O(k^N)$ に比べるとはるかに小さい。したがって完全パス数の多い回路に対しても短時間で機能的活性化不能パスを求めることができる。

部分経路活性化手法の概略を以下に示す。

Procedure:

1. 回路内の分岐枝から分岐幹までの部分経路を活性化させるための必要割当を求める
2. 各部分経路の必要割当を比較し、矛盾する必要割当を持つ部分経路の組合せを得る
3. 得られたそれぞれの部分経路の始点である分岐枝のペアを b-f ペアとする

b-f ペアの情報は単純なリストとして記録し、機能的活性化可能パスの抽出に用いる。

3 パス選択手法

3.1 機能的活性化可能な最長パスの抽出

本研究では、より有効なパス選択のアプローチとして、回路内の各信号線に対して、その信号線を含む最も長い機能的活性化可能パスの集合を求める考え方をとる。この戦略では、テストの対象となるパスは常にテスト可能であり、しかも各信号線を含む最も大きな遅延を与えるパスである。そのため、製造された回路内に複数存在する遅延を生じさせるような欠陥による故障を検出できる可能性が高い。本節では、このアプローチによるパス選択の実現のために、回路内の任意の信号線を含む最も長い機能的活性化可能パスを抽出する手続きを開発する。

以下では、信号線は遷移に関する情報をも含むものとする。b-f ペアの表が与えられたとき、回路内のある信号線 l を通る機能的活性化可能なパスは、外部入力から外部出力までの信号線を順次処理してゆくパス数え上げに類似した方法によって抽出可能である。外部入力から l までの部分論理パスは次の 3 通りに分類することができる。

- (i) b-line と、それに対応する f-line を含むパス
- (ii) b-line を含み、それに対応する f-line を l から外部出力までに含むパス
- (iii) 上記以外のパス

(i) のパスは外部入力から l までに b-f ペアを含んでおり、機能的活性化不能であることが確定したパスである。したがって (i) に分類されたパスはテスト不能である。(ii) のパスは外部入力から l までに既に b-line が含まれているパスである。これらのパスは、 l から外部出力までに f-line が出現して b-f ペアを構成し、(i) に分類される可能性がある。(iii) のパスは、外部入力から信号線 l に至るまでにいかなる b-line も含まないパスである。これらのパスは、b-line が出現したとき、(ii) に分類される。

例を図2に示す。 l_a, l_c が b-f ペアをなすと仮定する。 l_a におけるパスの分類では、 l_a 自身が b-line であるため、 l_a が (ii) に分類される。 l_b においては、(l_a, l_b) が (ii) に分類され、 l_d からのパスは (iii) に分類される。 l_c においては、 l_b において (ii) であったパスの中に l_c が f-line となるものがあるので、それが (i) に分類され、それ以外は (iii) に分類される。

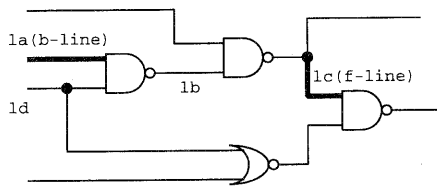


図 2: パスの分類

上述のパスの分類を、外部入力から外部出力まで順次行った結果、最終的に、機能的活性化可能な完全論理パスが外部出力で得られる。すなわち、外部出力において (iii) に分類されたパスが、機能的活性化可能な完全論理パスである。実際のインプリメンテーションでは、(ii) と (iii) のパスを信号線のリストで記録する。(iii) のパス集合は最長パスのみリストに記録する。(ii) のパス集合の計算には、パスが含む b-line に対応する f-line の情報が必要である。そのため、(ii) のパス集合の計算には複数のパス部分集合のリストと、各部分集合に付随する f-line のリストからなるデータ構造体を用いる。(i) のパス集合は分類される過程で逐次破棄する。

提案手法は、パスの分類と並行して蓄積遅延時間の計算を行う。蓄積遅延時間は、各信号線について、入力側信号線における蓄積遅延の最大値に信号線自体の遅延時間を加算することによって求められる。ここで、(iii) に分類されたパスのうち、蓄積遅延時間が最大でないも

のは機能的活性化可能な最長パスを構成しない。本手法ではこのようなパスはテストの対象として選択しない。そのため、各信号線における (iii) のパス集合の計算過程で、最長でないパスを破棄する。

ある信号線を通る全てのパスを分類する正確なアルゴリズムを図3に示す。

なお、図3において、 PP_x, PS, L はそれぞれ部分論理パス、部分論理パスの集合、および信号線の集合を表す変数である。 $SP(l)$ は外部入力から信号線 l までの (iii) のパスの集合である。 $PEND(l)$ はパスの集合と f-line の集合とのペアを要素として持ち、(ii) のパスの集合を表す。

3.2 パス選択のフレームワーク

ある信号線 l について、 l を通る機能的活性化可能な最長パスが、既に他の信号線を対象としたテストの対象となるパスとして抽出されているとき、 l はカバーされていると表記する。以下では全ての信号線をカバーする、より小さなパス集合を求める問題について述べる。

機能的活性化可能な最長パスを回路内の全ての信号線について抽出した場合、明らかに全ての信号線をカバーすることができる。しかしこの方法では、あるパスが複数の信号線をカバーする場合、その複数の信号線に対して同一のパスの抽出を繰り返すことになるため、結果として冗長な計算が発生する。そこで本研究では、これらの冗長な計算を前もって回避しつつ全ての信号線をカバーするパス集合を求めるフレームワークを開発する。

同一のパスによってカバーされる信号線にはいくつかの場合がある。例えば、回路の最長パスに含まれる信号線は、全てがその最長パスによってカバーされる。また、一入力一出力ゲートの入出力信号線は、同一のパスによってカバーされる。これと同様に、同じファンアウトフリーリージョン (FFR) に含まれる信号線には、同じパスによってカバーされるものがある。

図4に例を示す。信号線 l は、分岐の枝 l_{c1}, \dots, l_{c5} を入力とする各 FFR に含まれる。各始点から l までには分岐はないため、 l_{c1}, \dots, l_{c5} をカバーするパスの中で最も長いものが l を含む最も長いパスである。

この関係は全ての信号線について成り立つ。したがって、回路内に存在する全ての FFR の入力をカバーするパス集合を求めることにより、全ての信号線をカバーするパス集合を得ることができる。本研究で開発するフレームワークは、各 FFR の入力となる信号線に対し

```

draw_paths( line li )
{
  Mark all lines which can reach li;
  for each marked line lj {
    set SP(lj)={};
    set PEND(lj)={};
    /* propagating paths information */
    if lj is PI {
      if partial path PPx including lj is not FU
        add PPx=(lj) to SP(lj);
    }
    else {
      for each FanIn lk of lj {
        if SP(lk) is longer than SP(lj)
          set SP(lj)=SP(lk);
        add PEND(lk) to PEND(lj);
      }
      if partial path PPx including lj is FU {
        set PEND(lj)={};
        set SP(lj)={};
      }
      else {
        add lj to each PPx in SP(lj);
        for each ( PS, L ) pair in PEND(lj)
          add lj to each PPx in PS;
      }
    }

    /* remove shorter paths in PEND */
    for each ( PS, L ) pair in PEND(lj)
      if PS has shorter paths than SP(lj)
        remove ( PS, L ) pair from PEND(lj);

    /* check if lj is a f-line */
    for each ( PS, L ) pair in PEND(lj) {
      if lj is in L
        remove ( PS, L ) pair from PEND(lj);
      else
        for each lk in L {
          if lk never appears after lj
            remove lk from L;
          if L is empty {
            if PS is longer than SP(lj)
              set SP(lj)=PS;
            remove ( PS, L ) pair from PEND(lj);
          }
        }
    }

    /* check if lj is a b-line */
    if (lj is a b-line) and (corresponding f-line appears after lj) {
      add ( SP(lj), φ ) to PEND(lj);
      set SP(lj)={};
      for each ( PS, L ) pair in PEND(lj)
        add f-lines which corresponds to b-line of lj to L;
    }
  }
  LP={};
  for each PO lj {
    for each ( PS, L ) pair in PEND(lj)
      if PS is longer than SP(lj)
        set SP(lj)=PS;
    if SP(lj) is longer than LP
      LP = SP(lj);
  }
}
Functionally sensitizable longest paths set for li is LP.

```

図 3: パス抽出手順

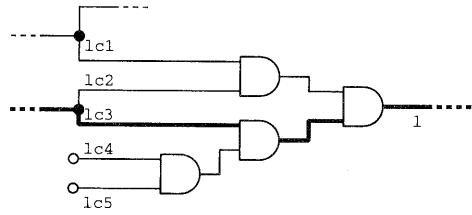


図 4: 分岐の枝によるカバー

て、最も長い機能的活性化可能パスを抽出する処理を基本とする。

パス選択の全体のフレームワークを以下に示す。

1. カバーされていない分岐の枝 l_i を選択
2. draw_paths(l_i) を実行
3. 全ての分岐の枝がカバーされていなければ 1 に戻る

4 実験結果

提案手法を SUN Ultra-1 ワークステーション Model 140(192MB) 上に C 言語でプログラム化し、IS-CAS'85 ベンチマーク回路に適用した。遅延割当として、全ての信号線にユニットディレイを仮定して実験を行った。プログラムのオプションとして、同じ長さの最長パスは全てテストの対象とするフレームワーク 1 と、同じ長さの最長パスのうち 1 本をテストの対象とするフレームワーク 2 とを用意した。

実験結果を表 1 に示す。第 1 列から順に、回路名、全論理パス数、部分経路活性化手法による機能的活性化可能パス数、フレームワーク 1 によるテスト対象パス数、フレームワーク 2 によるテスト対象パス数、フレームワーク 1 の実行時間、フレームワーク 2 の実行時間をそれぞれ示す。提案手法は記憶容量の不足により、c6288 に関しては計算を完了することが出来なかった。他の回路については実用的な時間内で計算を完了し、全ての回路に対してより小さなテスト対象パス集合を得られたことが示されている。

実験では遅延割当としてユニットディレイを仮定しているため、蓄積遅延時間が同一であるパスが多く、フレームワーク 1 とフレームワーク 2 との間で、選択されるパス数に大きな差が生じた。実際の回路における遅延

表 1: 実験結果

circuit	total paths	non-FU	framework 1	framework 2	CPU-time 1[s]	CPU-time 2[s]
c880	17284	17121	2197	729	1	1
c1355	8346432	1570272	60456	922	615	211
c1908	1458114	956101	4877	1293	149	111
c2670	1359920	165843	12873	1779	5	5
c3540	57353342	17901186	27351	2683	235	232
c5315	2682610	824445	6998	4247	9	9
c7552	1452988	567664	11908	5221	17	16

割当がパスごとに多様化する場合、選択されるパス数はフレームワーク 2 による実験結果に近づく。また、論理合成された回路等では遅延割当が多様化しないことが予想され、そのような回路ではフレームワーク 1 による実験結果に近づくと考えられる。

5 まとめ

本研究では、組合せ回路を対象に、より小さなテストすべきパス集合を選択する手法を提案した。実験結果は提案手法が実用的な時間内で機能的活性化可能な最長パスの集合を抽出できたことを示した。今後の課題として、提案手法によって抽出されたパスに対するテスト入力の生成があるが、抽出されたパスが機能的活性化可能であることは保証されていない。よって、抽出されたパスに関する機能的活性化可能性の検証と、それらがテスト不能である場合の対応とを併せて検討していく。

References:

- [Smit85] G.L.Smith, "Model for Delay Faults Based upon Paths," ITC-85, pp.342-349, 1985.
- [Barz83] Z.Barzilai and B.K.Rosen, "Comparison of AC Self-testing Procedures," ITC-83, pp.89-01, 1983.
- [Pome92] I.Pomeranz and S.M.Reddy, "An Efficient Non-Enumerative Method to Estimate Path Delay Fault Coverage," ICCAD-92, pp.560-567, 1992.
- [Chen93] K.-T.Cheng and H.-C.Chen, "Delay Testing For Non-Robust Untestable Circuits," ITC-93, pp.954-961, 1993.
- [Li89] W.-N.Li, S.M.Reddy, S.K.Sahni, "On Path Selection in Combinational Logic Circuits," IEEE Trans. on CAD., vol.8, pp.56-63, 1989.

[Spar95] U.Sparmann, D.Luxenburger, K.-T.Cheng, S.M.Reddy, "Fast Identification of Robust Dependent Path Delay Faults," DAC-95, pp.119-125, 1995.

[Lam93] W.K.Lam, A.Saldanha, R.K.Brayton, A.L.Sangiiovanni-Vincentelli, "Delay Fault Coverage, Test Set Size, and Performance Trade-Offs," IEEE Trans. on CAD, vol.14, pp.32-44, 1995.

[Kaji97] S.Kajihara, K.Kinoshita, I.Pomeranz, S.M.Reddy, "A Method for Identifying Robust Dependent and Functionally Unsensitizable Paths," International conf. on VLSI Design'97, pp.82-87, 1997.

[Li97] Z.Li, Y.Min, R.K.Brayton, "Efficient Identification of Non-Robustly Untestable Path Delay Faults," ITC-97, pp.992-997, 1997.

[Hera97] K.Heragu, J.H.Patel, V.D.Agrawal, "Fast Identification of Untestable Delay Faults Using Implications," ICCAD-97, pp.642-647, 1997.