

## 疑似ブール最適化を用いた故障診断容易化のためのコントローラの 状態遷移出力のドントケア割当て法

日大生産工(学部) ○大塚 裕衣 日大生産工(院) 千田 祐弥 日大生産工(院) 徐 浩豊

日大生産工 細川 利典 明治大学 山崎 浩二

### 1. はじめに

半導体微細化技術の進歩に伴い、超大規模集積回路 (Very Large Scale Integrated circuits: VLSI) において、異常動作の物理的な原因を特定する故障解析は、歩留まりの向上のために重要である。故障解析では、電子顕微鏡などを用いて故障 VLSI 内部の観測を行うため、多大なコストを要する。そのため、故障 VLSI に存在する可能性のある故障 (被疑故障) の数を事前にできる限り絞り込んでおく故障診断[1]が、故障解析コストの低減のために重要となる。故障診断では、故障 VLSI の異常な外部出力応答を裏付けることのできる故障箇所を推定する。

これまでに、縮退故障やブリッジ故障などの特定の故障モデルに対応した故障診断手法が数多く提案されている[2-6]。また、ゲートレベルやレイアウトレベルにおける観測ポイント挿入 (Observation Point Insertion : OPI) などの診断分解能を考慮した設計手法が提案されており、被疑故障数の削減が報告されている[7-9]。しかしながら、ゲートレベルやレイアウトレベルの回路は回路を構成する部品数が非常に多く、高速で効果的な個所に観測ポイントを挿入することが困難である。また、挿入された観測ポイントによる面積オーバーヘッドの増大や論理合成で行ったタイミングの最適性を損失するなどの課題が挙げられる。

本論文では、ゲートレベルやレイアウトレベルと比較して抽象度の高いレジスタ転送レベル (Register Transfer Level : RTL) に着目している。対象とする RTL 回路はデータパスとコントローラで構成され、データパスからコントローラへの状態信号とコントローラからデータパスへの制御信号によって接続されている。また、コントローラは有限状態機械で設計されていると仮定し、状態遷移時に制御信号値がデータパスに供給される。それらの制御信号値にはドントケア (X) が含まれる場合があり、論理合成時に面積の最小化を指向して、X に論理値が割当てられる。

文献[10]で、RTL における誤り経路追跡の結果を用いて被疑ハードウェア要素数が少なくなるように、コントローラの各状態遷移における制御信号

値 X に論理値を割当てる手法が提案されている。複数の観測点で誤りが観測される場合においては、被疑故障数が少なくなることが報告されている。しかしながら、全ての状態遷移において、観測点で誤りが観測されるケースを網羅していないため、観測点で誤りが観測されるケースが仮定されていないテスト結果においては、被疑ハードウェア要素数が増加する。そこで本論文では、RTL での診断分解能の向上のために、各状態遷移で識別できないハードウェア要素ペア数の最小化を指向し、制御信号の X に論理値を割当てる手法を提案する。本手法では、コントローラの状態遷移における制御信号の X 割当て問題を疑似ブール最適化問題で定式化した。

本手法では、フルスキャン設計を前提とし、フリップフロップ (FF) の D 端子である疑似外部出力を外部出力として扱う。また、論理故障を故障診断の故障モデルとした。

### 2. 諸定義

#### ●定義 1: 識別不能ハードウェア要素 (信号線) 集合

テスト集合  $T$  が与えられたとき、 $T$  で故障シミュレーションを実行し、検出された観測点集合が一致するハードウェア要素 (信号線) から構成される集合を識別不能ハードウェア要素 (信号線) 集合と呼ぶ。

#### ●定義 2: 状態遷移 $s$ で識別不可能なハードウェア要素ペア

状態遷移  $s$  で構造的記号シミュレーション[10]を実行した結果、テスト不可能なハードウェア要素ペア、またはテスト可能であるが観測点集合が一致するハードウェア要素ペアを  $s$  で識別不可能なハードウェア要素ペアという。また、 $s$  で識別不可能なハードウェア要素ペア以外のペアを  $s$  で識別可能なハードウェア要素ペアという。

#### ●定義 3: 状態遷移集合で識別不可能なハードウェア要素ペア

状態遷移集合  $st$  に属する状態遷移  $s, t$  において、構造的記号シミュレーションの結果、テスト不可

A Don't Care Filling Method of Outputs on Each State Transition for Controllers to Enhance Fault Diagnosability Using Pseudo Boolean Optimization

Yui OTSUKA, Yuya CHIDA, Haofeng XU, Toshinori HOSOKAWA, Kouji YAMAZAKI

能なハードウェア要素ペア, またはテスト可能であるが観測点集合が一致するハードウェア要素ペアを st で識別不可能なハードウェア要素ペアという. また, st で識別不可能なハードウェア要素ペア以外のペアを st で識別可能なハードウェア要素ペアという.

### 3. 制御信号値の X 割当て法

本章では, 各状態遷移における X 割当て法について述べる. 本手法では, 被疑ハードウェア要素削減のため, 各状態遷移で識別可能なハードウェア要素が増大するように, コントローラの制御信号値の X に論理値を割当てて.

図 1 に, X 割当て ST2-0 でテスト可能なハードウェア要素と X 割当て ST2-1 でテスト可能なハードウェア要素を示す. 図 1 において, R0, R1 はレジスタであり, テスト可能なハードウェア要素の観測点である. また, T はテスト可能なハードウェア要素の集合を示しており, T0, T1, T2 はそれぞれ, R0 のみでテスト可能なハードウェア要素集合, R1 のみでテスト可能なハードウェア要素集合, R0 と R1 の両方でテスト可能なハードウェア要素集合を表している. 図中の括弧内の数値はそれぞれ T0, T1, T2 のハードウェア要素数である. また, テスト不可能なハードウェア要素の集合は T3 とする. この例では全ハードウェア要素数を 20 とする. このとき, ST2-0 において, 集合 T0 のハードウェア要素数は 3, 集合 T1 のハードウェア要素数は 14, 集合 T2 のハードウェア要素数は 1, 集合 T3 のハードウェア要素数は 2 である. ここで, 識別不可能なハードウェア要素ペア数は, 集合 T0, T1, T2, T3 に属するハードウェア要素ペアの和である. 従って, ST2-0 の識別不可能なハードウェア要素ペア数は,  $3C_2 + 14C_2 + 1C_2$  を計算した 95 である. さらに, ST2-1 において, 集合 T0 のハードウェア要素数は 7, 集合 T1 のハードウェア要素数は 9, 集合 T2 のハードウェア要素数は 2, 集合 T3 のハードウェア要素数は 2 である. また, ST2-1 の識別不可能なハードウェア要素ペア数は,  $7C_2 + 9C_2 + 2C_2 + 2C_2$  を計算した 59 である.

識別可能なハードウェア要素ペア数は全ハードウェア要素ペア数から識別不可能なハードウェア要素ペア数を引くことで求めることができる. 全ハードウェア要素ペア数は 190 である. 従って, ST2-0 で識別可能なハードウェア要素ペア数は  $190 - 95$  を計算して 95, ST2-1 で識別可能なハードウェア要素ペア数は  $190 - 59$  を計算して 131 である. 本手法は, 各状態遷移で識別可能なハードウェア要素が増大するように X 割当てを行うため, この例では ST2-1 を選択する.

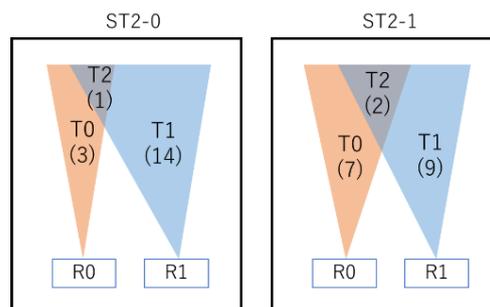


図 1. テスト可能なハードウェア要素

### 4. 問題定式化

本章では 3 章で述べたコントローラの各状態遷移における制御信号値の X 割当て問題を疑似ブール最適化問題 (Pseudo Boolean Optimization : PBO) で定式化する.

#### A. 定式化

$$\sum_{s=1}^M \sum_{i=1}^N \sum_{k=1}^P \sum_{h=k+1}^P W_{s-i-k-h} \cdot Y_{s-i} \quad \dots(1)$$

式(1)は前章で説明した制御信号値の X 割当て法を定式化したのものである. M は状態遷移数, N は状態遷移 s の X 割当ての場合の数, P は制御信号の故障を含む全てのハードウェア要素数を表す.  $W_{s-i-k-h}$  は状態遷移 s の X 割当て i でハードウェア要素 k, h が識別可能か否かを表す変数であり, 1 なら識別可能, 0 なら識別不可能であることを示す.  $Y_{s-i} (\in \{0,1\})$  は状態遷移 s の X 割当て i が選択されるか否かを表す変数であり, 1 なら選択され, 0 なら選択されないことを示す.  $Y_{s-i}$  は任意の状態遷移における識別可能なハードウェア要素数を増大することを目的とし, 式(1)を最大化する  $Y_{s-i}$  が選択される.

#### B. PBO の制約条件

$$\sum_{i=1}^N Y_{s-i} = 1 (\forall s) \quad \dots(2)$$

式(2)に, 各状態遷移において, 選択される X 割当てが 1 つであるという制約を示す. N は状態遷移 s の X 割当ての場合の数を表す.  $Y_{s-i} (\in \{0,1\})$  は状態遷移 s の X 割当て i が選択されるか否かを表す変数であり, 1 なら選択され, 0 なら選択されないことを示す.

$$\bigwedge_{k=1}^P \bigvee_{s=1}^M \bigvee_{i=1}^N Y_{s-i} \cdot Z_{s-i-k} = 1 \quad \dots(3)$$

式(3)に、全てのハードウェア要素が少なくとも1つの状態遷移でテスト可能であるという制約を示す。Pは制御信号の故障を含む全てのハードウェア要素数、Mは状態遷移数、Nは状態遷移sのX割当ての場合の数を表す。Y<sub>s-i</sub>(∈{0,1})は状態遷移sのX割当てiが選択されるか否かを表す変数であり、1なら選択され、0なら選択されないことを示す。また、Z<sub>s-i-k</sub>(∈{0,1})は状態遷移sのX割当てiでハードウェア要素kがテスト可能か否かを表す変数であり、1ならテスト可能、0ならテスト不可能であることを示す。式(3)は制御信号の全ての故障を含む全ハードウェア要素が少なくとも1つの状態遷移でテスト可能であることを保証することを目的としている。

$$\bigwedge_{k=1}^P \bigwedge_{h=k+1}^P \bigvee_{s=1}^M \bigvee_{i=1}^N W_{s-i-k-h} \cdot Y_{s-i} = 1 \quad \dots(4)$$

式(4)に、全てのハードウェア要素ペアが少なくとも1つの状態遷移で識別可能であるという制約を示す。Pは制御信号の故障を含む全てのハードウェア要素数、Mは状態遷移数、Nは状態遷移sのX割当ての場合の数を表す。W<sub>s-i-k-h</sub>は状態遷移sのX割当てiでハードウェア要素k,hが識別可能か否かを表す変数であり、1なら識別可能、0なら識別不可能であることを示す。Y<sub>s-i</sub>(∈{0,1})は状態遷移sのX割当てiが選択されるか否かを表す変数であり、1なら選択され、0なら選択されないことを示す。式(4)は制御信号の全ての故障を含む全てのハードウェア要素ペアが少なくとも1つの状態遷移で識別可能であることを保証することを目的としている。

### C. 診断容易化設計法の全体アルゴリズム

図2に、提案手法である診断容易化設計の全体アルゴリズムを示す。入力、RTLコントローラCとRTLデータパスDである。出力は、X割当てによって更新されたRTLコントローラACである。まず、状態遷移時にコントローラからデータパスに供給された制御信号値を用いて、データパスの構造的記号シミュレーション[10]を実行する(4行目から8行目まで)。このとき、制御信号値にXが存在する場合、全てのX割当ての組合せに対して構造的記号シミュレーションが実行される。従って、状態遷移sの制御信号値に含まれるXの数がn個の場

```

1.  Input : RTL controller C, and RTL data-path D
2.  Output : RTL controller AC with X-filling
3.  X-filling_Controller(C, D) {
4.    for (s=1; s<M; s++) {
5.      for (i=1; i<N; i++) {
6.        INFOs-i
           = Structural_Symbolic_Sim(D, STs-i)
7.      }
8.    }
9.    (ST1, ST2, ..., STM) = PBO (∀INFOs-i);
10.   AC = Controller_X-filling(C, ∀STs);
11.   return(AC);
12. }

```

図2. 提案手法のアルゴリズム

合、2<sup>n</sup>(=N)回の構造的記号シミュレーションを行う。その後、コントローラの各状態遷移におけるX割当て問題をPBOとして定式化して式(1)を解くことで、制御信号に存在するXに割当てる制御信号値を得ることができる(9行目)。最後に、各状態遷移のコントローラの制御信号にPBOで求めた論理値を割当てることで(10行目)、X割当て済みのACを得ることができる(11行目)。

### 5. 実験結果

本論文では、提案した故障容易化設計法を施した回路に対して、論理合成、スキーマ設計、縮退故障テスト生成を実行した後、識別不能ハードウェア要素集合を算出し評価した。実験は、RTLベンチマーク回路[11]を用いて行った。この実験では、32ビット幅のデータパスとコントローラで構成された回路を使用した。これらのRTL回路に、提案した診断容易化設計手法を適用し、状態遷移時の制御信号値のXに論理値を割当てた。PBOソルバーとしてClasp[12]を用いた。また、テストパターンについては、文献[13]を用いて生成し、生成したテスト集合を用いて識別不能ハードウェア要素集合を算出した。

表1は、ベンチマーク回路ex1における、故障診断の実験結果を示す。表1において、“circuit”は回路の名前、“#tp”はテストパターン数、“#fault”はハードウェア要素数を示している。また、“#unidentifiable fault sets”は識別不能ハードウェア要素集合数、“min”は識別不能ハードウェア要素集合の大きさの最小値、“max”は識別不能ハードウェア要素集合の大きさの最大値、“ave”は識別不能ハードウェア要素集合の大きさの平均値を示している。また、回路名において、“min”は最適化関数の値が最小化する割当てをした回路、“max”は最適化関数の値が最大となる割当てをした回路の結果を示している。

表 1. 実験結果

circuit	#tp	#fault	#unidentifiable fault sets	size				variance
				min	max	mode	ave	
ex1_min	189	4678	2376	1	5	1	1.97	0.96
ex1_max	159	4680	2373	1	7	1	1.97	0.98

回路” ex1\_min” と比較して” ex1\_max” が、識別不能ハードウェア要素集合数が 3 個少なく、識別不能ハードウェア要素集合の最大のサイズが 2 個大きくなった。従って、本手法は回路 ex1 に対して有効でないと言える。

## 6. まとめ

本論文では、コントローラの状態遷移の X 割当て法を提案し、制御信号値の X 割当て問題を PBO として定式化した。実験の結果、本手法は回路 ex1 において有効でないことが分かった。今後の課題としては、PBO 式の改善や他回路における本手法の故障診断への有効性の検証を行うことなどが挙げられる。

## 参考文献

[1] H. Y. Chang, E. Manning and G. Metzger: "Fault Diagnosis of Digital Systems", John Wiley & Sons, Inc. 1970.

[2] V. Boppana and W. K. Fuchs, "Fault dictionary compaction by output sequence removal," Proc. ICCAD, pp. 287-296, 2001.

[3] S. D. Millman, E. J. McCluskey and J. M. Acken, "Diagnosing CMOS Bridging Faults with Stuck-at Fault Dictionaries," Proc. ITC, pp. 860-870, 1990.

[4] S. Venkataraman and S. B. Drummonds, "A technique for logic fault diagnosis of interconnect open defects," Proc. VTS, pp. 313-318, 2000.

[5] I. Hartanto, S. Venkataraman, W. K. Fuchs, E. M. Rudnick, J. H. Patel, S. Chakravarty, "Diagnostic simulation of stuck-at faults in sequential circuits using compact lists," ACM Transactions on Design Automation of Electronic Systems (TODAES), pp. 471-489, Volume 6, Issue 4, October 2001.

[6] Y. Benabboud, A. Bosio, L. Dilillo, P. Girard, S. Pravossoudovitch, A. Virazel, and O. Riewer, "Delay Fault Diagnosis in Sequential Circuits," Proc. ATS, pp. 355-360, 2009.

[7] T. Yamada, K. Yamazaki, and E. J. McCluskey,

"A Simple Technique for Locating Gate-Level Faults in Combinational Circuits," Proc. ATS, pp. 65-70, 1995.

[8] I. Pomeranz, S. Venkataraman, and S. M. Reddy: "Z-DFD: Design-for-Diagnosability Based on the Concept of Z-detection," Proc. ITC, pp. 489-497, October 2004.

[9] N. Kuji, T. Ishihara, and S. Nakajima, "EB-Testing-PAD Method and Its Evaluation by Actual Devices," IEICE Trans. Inf. & Syst., Vol. E85-D, No10, pp. 1558-1563, October 2002.

[10] K. Tsuchibuchi, T. Hosokawa and K. Yamazaki, "A Don't Care Filling Method for Control Signal Values of Controllers to Enhance Fault Diagnosability at Register Transfer Level," The 22nd IEEE Workshop on RTL and High Level Testing, November 25-26, 2021.

[11] N. Toyota, X. Wen, S. Kajihara, and M. Sanada: "Quantifying Observability for Fault Diagnosis of VLSI Circuits," IEE 6th Workshop on RTL and High-Level Testing, Harbin, China, July 20-21, 2005.

[12] M. T.-C. Lee, "High-Level Test Synthesis of Digital VLSI Circuits", Artech House Publishers, 1997.

[13] M. Henftling, H. C. Wittmann, and K. J. Antreich, "A Single-Path-Oriented Fault Effect Propagation in Digital Circuits Considering Multiple-Path Sensitization," Proc. 1995 IEEE/ACM International Conference on Computer-Aided Design, pp. 304-309, 1995.