

# レジスタ転送レベルにおける識別可能故障ペア推定数最大化のための制御信号のドントケア割当て手法

日大生産工(院) ○大塚 裕衣 日大生産工 細川 利典  
京都産業大 吉村 正義 明治大 山崎 浩二

## 1. まえがき

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おり[2-4]、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などの診断分解能向上を考慮した設計手法が提案されており、被疑故障数が削減できることが報告されている[5-7]。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、高速に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。

その課題を解決するために、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) における診断容易化設計手法が提案されている[8-10]。対象とする RTL 回路はデータバスとコントローラで構成され、データバ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バ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バスに供給される。それらの制御信号値にはドントケア(don't care: X)が含まれる場合があり、通常は論理合成時に面積の削減を指向して、X に論理値が割当てられる。また本論文では、スキューン設計を前提としているため、レジスタの入力信号線と外部出力は区別せずに観測点と呼ぶ。

文献[10]では、各状態遷移における識別可能な重み付きハードウェア要素ペア数の最大化を指向したコントローラの X 割当て手法が提案されている。この手法では、各ハードウェア要素にその面積を重みとして付与している。しかしながら、マルチプレクサ (multiplexer: MUX) において、制御信号値によってゲートレベルにおける MUX のテスト可能な信号

線が異なることを考慮できていなかったため、分解能を正確に推定することができていない。また、文献[11]の解析において、RTL においてテスト可能でかつ識別不能となったハードウェア要素ペアに対応するゲートレベルの故障ペアは平均で 95%識別可能となることが示されている。そのため本論文では、重みを故障数に変更し、RTL で識別可能となったハードウェア要素ペアに加えて、RTL でテスト可能な観測点が一一致したために識別不能となったハードウェア要素ペアの重みも計算に含めることとする。

本論文では、全状態遷移を通して識別可能故障ペア推定数が増大するような X 割当て法を提案する。全ハードウェア要素のテスト実行回数が増加する X 割当てに焦点を当て、その X 割当ての中から全状態遷移を通して識別可能故障ペア推定数が最大化するように制御信号の X に論理値を割当てる手法を提案する。本論文では、文献[9-10]と同様に、コントローラの状態遷移における制御信号の X 割当て問題を擬似ブール最適化問題[12]で定式化する。また、対象回路はフルスキューン設計を適用することを前提とし、論理故障を故障診断の故障モデルとした。さらに、コントローラの制御信号の X の総数が多い回路に対して X 割当てを行うためのヒューリスティックアルゴリズムも提案する。

## 2. テスト可能なハードウェア要素

本論文では、データバス内のモジュールの入出力ポートと制御信号線の故障をハードウェア要素と呼ぶ。RTL 回路における状態遷移  $st$  でテスト可能なハードウェア要素は構造的記号シミュレーション[8]によって算出され、あるハードウェア要素がテスト可能となった状態遷移の数をテスト実行回数と呼ぶ。

### ●構造的記号シミュレーション

構造的記号シミュレーションとは、RTL 回路のデータバスのハードウェア要素のテスト可能性を解析する手法であり、各状態遷移でテスト可能なハードウェア要素の列挙が可能である。制御信号値によって、外部入力やレジスタから制御可能なハードウェア要素に C シンボル[13]、観測点で観測可能なハードウェア要素に O シンボル[13]を割当て、C シンボルと O シンボルが割当てられたハードウェア要素をテスト可能なハードウェア要素としている。また、構造的記号シミュレーションには楽観シミュレーション[13]と悲観シミュレーション[13]がある。制御信号値に X が存在した場合、楽観構造的記号シミュレーション[13]ではハードウェア要素がテスト可能と

An X-Filling Method of Control Signals to Maximize the Estimated Number of Distinguishable Fault Pairs at Register Transfer Level

Yui OTSUKA, Toshinori HOSOKAWA, Masayoshi YOSHIMURA, Kouji YAMAZAKI

なる論理値が、悲観構造的記号シミュレーション[13]ではハードウェア要素がテスト不能となる論理値が割当てられると仮定してシミュレーションを行う。

### 3. 識別可能故障ペア推定数

識別可能故障ペア推定数とは、対象 RTL 回路を論理合成することで生成されるゲートレベル回路において識別可能となる可能性のある故障ペア数の推定値である。また、各モジュールの出力ポートと MUX の入出力ポート、制御信号ポートを使用し、使用するハードウェア要素には対応するゲートレベルの故障数を重みとして付与している。

文献[11]によって RTL で識別可能となったハードウェア要素ペアに対応するゲートレベルの故障ペアは 100%識別可能であることがわかっている。また、RTL でテスト不能で識別不能となったハードウェア要素ペアはゲートレベルで一切識別されないが、RTL でテスト可能で識別不能となったハードウェア要素ペアはゲートレベルにおいて平均で 95%識別可能であることがわかっている。ゲートレベルで各故障ペアを識別するには、少なくとも 1 つの状態遷移で識別可能となれば良い。したがって、全状態遷移を通して少なくとも 1 つの状態遷移で識別可能となったハードウェア要素ペアに対しては各ハードウェア要素の重みの積を計算する。さらに、全状態遷移で識別可能とはならなかったが少なくとも 1 つの状態遷移でテスト可能な観測点が一致したために識別不能となったハードウェア要素ペアに対しては各ハードウェア要素ペアの重みの積×0.95 を計算する。各ハードウェア要素ペアの計算結果の総和を識別可能故障ペア推定数とする。

### 4. ヒューリスティックアルゴリズム

本手法では、各ハードウェア要素のテスト実行回数を増加させることで X 割当ての選択肢を限定するとともに、識別可能故障ペア推定数を増加させることを目的としている。テスト実行回数を増加させたいハードウェア要素を対象として X 割当てを行う。

#### 4-1. X 割当て法

各状態遷移において対象ハードウェア要素がテスト可能となる X 割当てを算出し、算出した全ての X 割当ての中から識別可能故障ペア推定数が増大するように各状態遷移の X 割当てを選択する。

図 1(a)のデータパスと図 1(b)のコントローラで構成されている RTL 回路を例に説明する。例えば、add0\_output, mux1\_input1, mux2\_input0, mux2\_input1 のテスト実行回数を増加させる X 割当てを表 1 に示す。各行の ST は状態遷移を表しており、説明のために各 X 割当てに変数  $x_i$  ( $i$  は自然数) を割当てている。列 orig は X 割当て前のコントローラの制御信号値を示しており、その右列は各状態遷移において各対象ハードウェア要素がある 1 つの観測点で観測可能となる X 割当てを示している。さらに、

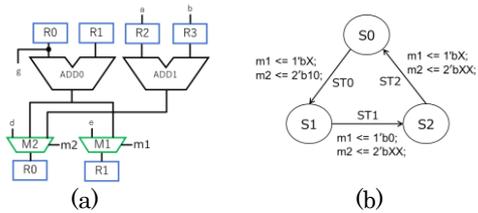


図 1. データパスとコントローラ

	orig	各ハードウェア要素が テスト可能となる X 割当て	マージによって算出された X 割当て
ST0	<x1> m1 <= 1'bX; m2 <= 2'b10;	<x2> m1 <= 1'b1; m2 <= 2'b10;	
ST1	<x3> m1 <= 1'b0; m2 <= 2'bXX;	<x4> m1 <= 1'b0; m2 <= 2'b01;	<x5> m1 <= 1'b0; m2 <= 2'b00;
ST2	<x6> m1 <= 1'bX; m2 <= 2'bXX;	<x7> m1 <= 1'b1; m2 <= 2'bXX;	<x8> m1 <= 1'bX; m2 <= 2'b00;
		<x9> m1 <= 1'bX; m2 <= 2'b01;	<x10> m1 <= 1'b1; m2 <= 2'b00;
			<x11> m1 <= 1'b1; m2 <= 2'b01;

表 1. X 割当て一覧

算出した X 割当てに X が存在した場合、他の X 割当てとマージできる場合がある。複数の X 割当てをマージすることで、識別可能故障ペア推定数が増加する X 割当てを算出することができる可能性があり、さらに 1 つの X 割当てで複数の対象ハードウェア要素のテスト実行回数を増加させることができる可能性がある。そのため、値が衝突しない全ての X 割当ての組合せに対してマージを行う。この例におけるマージ結果は表 1 の左端の列に示しており、x7 と x8, x7 と x9 をマージして X 割当て x10, x11 が算出されている。この例で列挙できた X 割当ては 11 個であり、対象ハードウェア要素のテスト実行回数を増加させるとともに識別可能故障ペア推定数を増大させる X 割当てを選択する。

#### 4-2. 擬似ブール最適化問題

本節では第 1 節で述べたコントローラの X 割当て問題を擬似ブール最適化問題(Pseudo Boolean Optimization: PBO)で定式化する。

##### 4-2-1. 定式化

$$\sum_{p=1}^Q W_p \times (X_p \sim Y_p \times 0.95 + Y_p \times 1.0) \quad \dots (1)$$

式(1)は前節で説明した制御信号値の X 割当て法を定式化したものである。Q は全ハードウェア要素ペア数、 $W_p$  はハードウェア要素ペア p の重みを表す。また、 $X_p \in \{0,1\}$  は選択された X 割当てでハードウェア要素ペア p がテスト可能な観測点が一致したために識別不能となる(1)か否(0)かを表し、 $Y_p \in \{0,1\}$  はハードウェア要素ペア p が選択された X 割当てで識別可能となった(1)か否(0)かを表す。識別可能故障ペア推定数を増大することを目的とし、式(1)を最大化する  $X_p$  と  $Y_p$  が決定される。

##### 4-2-2. PBO の制約条件

$$\sum_{k=1}^{K_i} z_{ik} = 1 \quad (\forall i) \quad \dots (2)$$

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。 $K_i$  は状態遷移 i の X 割当て数、 $z_{ik} \in \{0,1\}$  は状態遷移 i の k 番目の X 割当て  $z_{ik}$  が選択される(1)か否(0)かを表す。

$$\bigwedge_{i=1}^M \bigwedge_{k=1}^{K_i} z_{ik} \cdot T_{ijk} = 1 \quad (\forall j) \quad \dots (3)$$

式(3)に、全対象ハードウェア要素のテスト実行回数が増加するという制約を示す。\$M\$は状態遷移数、\$K\_i\$は状態遷移 \$i\$ の X 割当て数を表す。\$z\_{ik} \in \{0,1\}\$ は状態遷移 \$i\$ の \$k\$ 番目の X 割当て \$z\_{ik}\$ が選択される(1)か否(0)かを表し、\$T\_{ijk} \in \{0,1\}\$ は状態遷移 \$i\$ の \$k\$ 番目の X 割当てでハードウェア要素 \$j\$ がテスト可能(1)か否(0)かを表す。式(3)は全対象ハードウェア要素が少なくとも 1 回はテスト実行回数が増加することを保証することを目的としており、この制約を満たす解が存在しない場合はこの式は使用しない。

$$X_p \vee \bigwedge_{i=1}^M \bigwedge_{k=1}^{K_i} z_{ik} \cdot U_{ikp} = 0 \quad (\forall p) \quad \dots (4)$$

$$X_p \wedge \bigwedge_{i=1}^M \bigwedge_{k=1}^{K_i} z_{ik} \cdot U_{ikp} = 1 \quad (\forall p) \quad \dots (5)$$

式(4)(5)は、ハードウェア要素ペア \$p\$ がテスト可能でかつ識別不能となる X 割当てが選択されたか否かを表す式であり、この式は必ず片方のみを満たす必要がある。\$X\_p \in \{0,1\}\$ は選択された X 割当てでハードウェア要素ペア \$p\$ がテスト可能な観測点が一致したために識別不能となる(1)か否(0)かを表し、\$M\$は状態遷移数、\$K\_i\$は状態遷移 \$i\$ の X 割当て数を表す。\$z\_{ik} \in \{0,1\}\$ は状態遷移 \$i\$ の \$k\$ 番目の X 割当て \$z\_{ik}\$ が選択される(1)か否(0)かを表し、\$U\_{ikp}\$ は状態遷移 \$i\$ の \$k\$ 番目の X 割当てでハードウェア要素ペア \$p\$ がテスト可能でかつ識別不能となる(1)か否(0)かを表す。

$$Y_p \vee \bigwedge_{i=1}^M \bigwedge_{k=1}^{K_i} z_{ik} \cdot D_{ikp} = 0 \quad (\forall p) \quad \dots (6)$$

$$Y_p \wedge \bigwedge_{i=1}^M \bigwedge_{k=1}^{K_i} z_{ik} \cdot D_{ikp} = 1 \quad (\forall p) \quad \dots (7)$$

式(6)(7)は、選択された X 割当てでハードウェア要素ペア \$p\$ が識別可能となるような X 割当てが選択されたか否かを表す式であり、この式は必ず片方のみを満たす必要がある。\$Y\_p \in \{0,1\}\$ はハードウェア要素ペア \$p\$ が選択された X 割当てで識別可能となる(1)か否(0)かを表し、\$M\$は状態遷移数、\$K\_i\$は状態遷移 \$i\$ の X 割当て数を表す。\$z\_{ik} \in \{0,1\}\$ は状態遷移 \$i\$ の \$k\$ 番目の X 割当て \$z\_{ik}\$ が選択される(1)か否(0)かを表し、\$D\_{ikp}\$ は状態遷移 \$i\$ の \$k\$ 番目の X 割当てでハードウェア要素ペア \$p\$ が識別可能となる(1)か否(0)かを表す。

```

1. Input : RTL controller C, RTL data-path D and Target hardware T
2. Output : RTL controller AC with X-filling
3. X-filling_Controller(C, D, T) {
4.   for (i=1; i<=M; i++) {
5.     FXi = fxorg;
6.     for (j=1; j<=|Tj|; j++) {
7.       FXi = FXi ∪ enumerate(i, j);
8.     }
9.     FXi = FXi ∪ merge(FXi);
10.  }
11. }
12. (MST1, MST2, ..., MSTM) = PBO(C, D, FX1, FX2, ..., FXM);
13. AC = augment(C, MST1, MST2, ..., MSTM);
14. return(AC);
15. }

```

図2. X割当て法の全体アルゴリズム

#### 4-2-3. X割当て法の全体アルゴリズム

図2に、X割当て法のアルゴリズムを示す。入力、RTLコントローラ \$C\$、RTLデータパス \$D\$、対象ハードウェア要素集合 \$T\$ 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。また、\$FX\_i\$ とは状態遷移 \$ST\_i\$ における X 割当て集合であり、\$fx\_{org}\$ とは状態遷移 \$ST\_i\$ のオリジナルの X 割当てである。\$MST\_i\$ は PBO が選択した状態遷移 \$ST\_i\$ の X 割当てを行って変更された状態遷移を表している。まず、状態遷移 \$ST\_i\$ の X 割当て集合を状態遷移 \$ST\_i\$ のオリジナルの X 割当てで初期化する(5行目)。次に、状態遷移 \$ST\_i\$ で対象ハードウェア要素 \$j\$ が1つの観測点で観測可能となるための X 割当てを列挙し(7行目)、求めた X 割当ての中で衝突しない X 割当てをマージした X 割当てを全て列挙する(10行目)。さらに、求めた X 割当てから PBO として式(1)を解くことで、制御信号の X に割当てる値を得ることができる(12行目)。その後、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、X 割当て済みのコントローラを得る(13行目)。最後に、X 割当て済みのコントローラを返却する(14行目)。

#### 5. インクリメンタルX割当て

本章では、第4章で提案した X 割当てを用いて段階的に各ハードウェア要素のテスト実行回数を増加させ、識別可能故障ペア推定数を最大化するアルゴリズムを提案する。

テスト実行回数を増加させることで、各ハードウェア要素の対応するゲートレベルの故障のペアが識別可能になる可能性を向上させることができる。また、偏りなく全ハードウェア要素のテスト実行回数を増やすことが分解能の向上に重要であると考えた。図3に、インクリメンタル X 割当てのアルゴリズムを示す。入力は RTL コントローラ \$C\$ と RTL データパス \$D\$ であり、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。また、ハードウェア要素 \$j\$ の悲観的なテスト実行回数を \$P\\_TE\_j\$、楽観的なテスト実行回数を \$O\\_TE\_j\$ で表している。また、\$THW\$ は対象ハードウェア要素集合、\$r\$ は対象とするハードウェア要素の悲観的なテスト実行回数、\$Num\\_X\$ はコントローラの制御信号に存在する X の数を表している。まず、悲観構造的記号シミュレーション(5行目)と楽観構造的記号シミュレーション(6行目)を実行して各ハードウェア要素の悲観と楽観のテスト実行

```

1. Input : RTL controller C and RTL data-path D
2. Output : RTL controller AC with X-filling
3. Heuristic_Algorithm(C, D) {
4.   do {
5.     (P\_TE1, P\_TE2, ..., P\_TEj) = Pessi_Structural_Symbol_Simulation(C, D);
6.     (O\_TE1, O\_TE2, ..., O\_TEj) = Opti_Structural_Symbol_Simulation(C, D);
7.     THW = Decision_Target_Hardware(P\_TE1, P\_TE2, ..., P\_TEj, O\_TE1, O\_TE2, ..., O\_TEj, r);
8.     if (THW != ∅) {
9.       C = X-filling_Controller(C, D, THW);
10.      r++;
11.     }
12.   } while (THW != ∅);
13.   AC = C;
14.   rreturn(AC);
15. }

```

図3. インクリメンタル X 割当てのアルゴリズム

回数を求める。その結果から、悲観的なテスト実行回数が  $r$ 、楽観的なテスト実行回数が  $r+1$  以上のハードウェア要素を  $X$  割当ての対象ハードウェア要素として抽出する (9 行目)。このとき、対象ハードウェア要素集合が空集合でない場合は、前章で説明した  $X$  割当てを行ってコントローラを更新する (13 行目)。その後、さらにテスト実行回数を増加させるために  $k$  の値をインクリメントする (14 行目)。5~15 行目の処理を対象ハードウェア要素集合が空集合となるまで繰り返し、その後  $X$  割当ての結果を  $X$  割当て済みのコントローラとし (17 行目)、 $X$  割当て済みのコントローラを返却する (18 行目)。

## 6. 実験結果

本論文では、提案した故障診断容易化設計法について RTL における識別可能故障ペア推定数を算出して評価した。本実験では、RTL ベンチマーク回路[14]のうち ex1,ex2,ex4,Tseng に対して提案したヒューリスティックアルゴリズムを適用し、PBO ソルバーとして Clasp[15]を用いた。ex1,ex4,Tseng においては全状態遷移を通してテスト不能なハードウェア要素が存在しなかったため、テスト実行回数を 0 回からそれ以上に増加させる  $X$  割当ては行っていない。また、すべての回路でテスト実行回数を 3 以上に増加させたところで楽観的と悲観的のテスト実行回数の平均値が同じとなったため、 $r=2$  で  $X$  割当てを終了した。また、評価対象として  $X$  にランダムな論理値を割当てたコントローラを作成した。ex1 においてはコントローラの  $X$  の総数が少ないため  $X$  に割当てられる論理値のすべての組合わせのコントローラを作成し、ex2,ex4,Tseng に対しては  $X$  にランダムな論理値を割当てたコントローラを 10,000 個作成した。作成した各コントローラとヒューリスティックによって求めたコントローラの識別可能故障ペア推定数の比較結果を図 4 に示す。図 4 において、緑がランダムに生成したコントローラ、橙色が  $X$  割当て手法を適用したコントローラの識別可能故障ペア推定数である。また、orig は  $X$  割当て前のコントローラを示している。図 4 から、どの回路においても  $X$  割当てによって識別可能故障ペア推定数が増大する  $X$  割当てを確実に求めることができていることがわかる。

## 7. まとめ

本論文では、全状態遷移を通じた識別可能故障ペア推定数を最大化するために、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。また、大規模回路への適用のためのヒューリスティックアルゴリズムも提案した。実験の結果、全ての回路で確実に識別可能故障ペア推定数が増加する  $X$  割当てを求めることができることがわかった。今後の課題としては、識別可能故障ペア推定数とゲートレベルにおける分解能の相関関係を解析することが挙げられる。

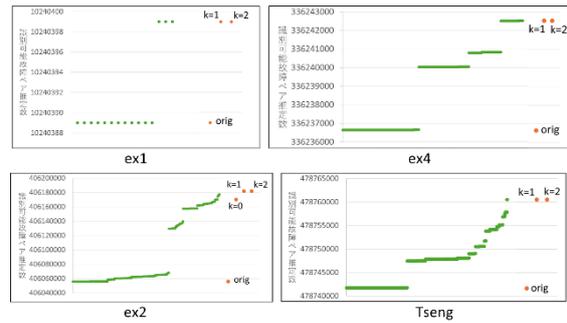


図 4. 識別可能故障ペア推定数の比較

## 参考文献

- 1) H.Y.Chang, E.Manning and G.Metze: "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) 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.
- 6) 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.
- 7) 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.
- 8) 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.
- 9) 大塚裕衣, 千田祐弥, 徐浩豊, 細川利典, 山崎浩二, "識別可能ハードウェア要素ペア数最大化のためのコントローラの制御信号のドントケア割当て法," 信学技報, vol. 122, no. 285, DC2022-41, pp. 37-42, November 2022.
- 10) 大塚裕衣, 千田祐弥, 徐浩豊, 細川利典, 山崎浩二, "識別可能重み付きハードウェア要素ペア数最大化のためのコントローラの制御信号のドントケア割当て法," 信学技報, vol. 123, no. 146, DC2023-12, pp. 25-30, August 2023.
- 11) 大塚裕衣, "状態遷移の識別不能ハードウェア要素ペアにおける識別可能故障ペア数の解析," 日本大学生産工学部学術講演会講演概要, 56th, ROMBUNNO.2-34, 2023.
- 12) Vasco Manquinho, Ruben Martins, and In'es Lynce, "Improving Unsatisfiability-Based Algorithms for Boolean Optimization", Theory and Applications of Satisfiability Testing-SAT 2010.
- 13) 徐浩豊, 細川利典, 吉村正義, 新井雅之 "並列テストのためのコントローラの制御信号のドントケア割当てアルゴリズム, " 信学技報, vol.122, no.205, DC2022-24, pp.37-42, October 2022.
- 14) 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.
- 15) M.T.-C.Lee, "High-Level Test Synthesis of Digital VLSI Circuits", Artech House Publishers, 1997.