

ダミー演算に基づく

診断容易化レジスタバインディングアルゴリズム

日大生産工 (院) ○久保倉 修司
 日大生産工 細川 利典
 明治大・情報コミュニケーション 山崎 浩二
 京都産業大・コンピュータ理工 吉村 正義

1. まえがき

超大規模集積回路 (Very Large Scale Integrated Circuits : VLSI) のテスト時に異常動作が観測された場合に物理的な原因を特定する故障解析 [1] が行われる。故障解析は、歩留まりの向上のために重要であり、近年の VLSI の大規模化、高集積化に伴い、電子顕微鏡などを用いて故障 VLSI 内部を観測するため、多大なコストを要する。そのため、故障 VLSI 内部に存在する可能性のある故障を事前に絞り込んでおく故障診断 [2] が、故障解析コストの低減のために重要となる。故障診断の実施により、故障 VLSI 内部の故障個所を推定した被疑故障集合が得られ、被疑故障集合中に実際の欠陥が存在し、被疑故障集合の要素数が少ないことが診断分解能の向上へつながる。

診断分解能を向上させる手法として 2 種類の手法が提案されている。1 つ目は、高診断分解能を指向したテスト生成法 [3][4] である。故障診断では、テスト集合を用いて故障個所を推定するため、テスト集合の診断品質が重要となる。文献 [3,4] ではテスト集合の診断品質として、識別可能な故障ペア数を用いている。

診断分解能を向上させる 2 つ目の手法として、レジスタ転送レベル (Register Transfer Level : RTL) における故障診断容易化設計 (Design for Diagnosability : DFD) 手法 [5][6] が提案されている。文献 [5,6] では、抽象度の高いレジスタ転送レベル (Register Transfer Level : RTL) に着目した DFD 手法が提案されている。RTL 回路はデータバスとコントローラから成り、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が互いに接続されているものを対象としている。コントローラは有限状態機械 (Finite State Machine : FSM) で設計されていると仮定し、状態遷移時に制御信号がデータバスに供給される。それらの制御信号値には多数のドントケア (X) が含まれており、設計の自由度が存在する。一般に、論理合成時には小面積化や低消費電力化を指向して X に論理値を割当るが、文献 [6] では各状態遷移においてデータバス中のハードウェア要素ペアが識別可能である

ことが、診断分解能の向上に重要であると考え、各状態遷移の X に対して識別可能故障ペア推定数を最大化するように論理値を割当てる手法が提案されている。

また、より抽象度の高い高位合成 (High-Level Synthesis : HLS) において DFD を考慮することは、RTL での X 割当てなどの回路構造を変更しない DFD 手法と異なり、回路構造を変更することができるため、RTL より高い診断容易化設計の自由度が存在するとして、診断容易化レジスタバインディング手法 [7] が提案されている。文献 [7] では、スキャン設計回路を対象とした RTL 回路において、診断容易な回路の特徴を解析し、その特徴を HLS において実現する手法を提案している。

また、非スキャン設計を前提とした高位合成におけるテスト容易化合成手法として、ダミー演算 [8] を使用した並列テスト法 [8] が提案されている。文献 [8] では、アイドル状態のハードウェアを活用し、ダミー演算を挿入することにより、機能動作と並列して、機能動作を行わないダミー演算に対応する演算器をテストし、テスト実行時間を削減している。

本論文では文献 [7] の手法に、文献 [8] のダミー演算の考え方を取り入れ、ダミー演算に基づく診断容易化レジスタバインディング手法を提案する。

本論文の構成は以下のとおりである。第 2 章では前提知識を説明し、第 3 章では提案手法を説明する。第 4 章では実験結果を示し、第 5 章では考察を述べる。

2. 前提知識

本章では、本論文で取り扱う用語を定義する。

定義 1：ダミー演算

バインディング済みデータフローグラフ (Binded Data-Flow Graph : BDFG) において、特定の時刻 t にアイドル状態演算器 A があり、かつ、時刻 t にアイドル状態レジスタ R があるとき、時刻 t において演算器 A はレジスタ R でダミー演算可能であるという。

定義 2：ハードウェア要素ペア

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び、それらのペアを文献[6]ではハードウェア要素ペアと定義している。これらペアを、4 タイプへ分類する。

(タイプ 1: 異なる観測点集合をもつテスト可能なハードウェア要素のペア)

両方のハードウェア要素がテスト可能であり、かつ、少なくとも 1 つの状態遷移において観測点集合が異なる場合にそのハードウェア要素ペアはタイプ 1 に分類される。

(タイプ 2: 一方がテスト可能で、もう一方がテスト不能なハードウェア要素のペア)

タイプ 1 以外で、少なくとも 1 つの状態遷移において、一方がテスト可能で、もう一方がテスト不能な場合にハードウェア要素ペアはタイプ 2 に分類される。

(タイプ 3: 同じ観測点集合をもつテスト可能なハードウェア要素のペア)

タイプ 1 やタイプ 2 に当てはまらず、1 つ以上の状態遷移において両方のハードウェア要素がテスト可能で、かつ同じ観測点集合を持つ場合にハードウェア要素ペアはタイプ 3 に分類される。

(タイプ 4: 両方テスト不能なハードウェア要素のペア)

すべての状態遷移において両方のハードウェア要素がテスト不能である場合にハードウェア要素ペアはタイプ 4 に分類される。

タイプ 1 及びタイプ 2 は識別可能ハードウェア要素ペアと呼び、タイプ 3 及びタイプ 4 は識別不能ハードウェア要素ペアと呼ぶ。

3. ダミー演算に基づく診断容易化レジスタバインディングアルゴリズム

本章では、ダミー演算に基づく診断容易化レジスタバインディング手法を提案する。

図 2 に、ダミー演算に基づく診断容易化レジスタバインディング手法の全体アルゴリズムを示す。

本手法では、レジスタバインディング時にレフトエッジアルゴリズムを適用して得られたレジスタ数を制約として与えるため、最初の動作としてレフトエッジバインディングを行う(1~2 行目)。続いて、各レジスタに対して共有可能な変数の組合せをすべて列挙する。ここでは制約として、同じ演算器から出力される変数はレジスタを共有不能としている。この制約は、後の演算器の観測レジスタ数最大化のために設けている(3 行目)。列挙された各レジスタに共有可能な変数の組合せから、レジスタに格納される変数が重複していない組合せを列挙し、診断容易 BDFG 候補集合 $BDFG_{set}$ を作成する(4 行目)。演算器の観測レジスタ数最大化を指向するため、 $BDFG_{set}$ の中から 1 つの BDFG を選択し($BDFG_i$)、演算器の観測レジスタ数が最大となっている

```

Input :  $F_{DFG}$  (演算器のみバインディング済みDFG)
Output : BestBDFGset(診断容易化レジスタバインディング済みBDFG集合)

1 :  $Fb_{DFG} = Left Edge Binding(F_{DFG});$ 
2 :  $Cons_{Reg} = Create constraints of register(Fb_{DFG});$ 
3 :  $RegList = Create All Combination varible in register (VarialbeList);$ 
4 :  $BDFG_{set} = Create BDFG(RegList);$ 
5 :  $Maximize Func OBP(BDFG_{set}, RO_{set});$ 
6 :  $Maximize Num Dummy calc(RO_{set}, Dummy_{set});$ 
7 :
8 :  $return Dummy_{set};$ 

1 : Maximize Func OBP(BDFG_{set}, RO_{set}){
2 : While(BDFG_{set} != NULL){
3 :    $tmp_{BDFG} = Select BDFG(BDFG_{set});$ 
4 :    $BDFG_{set} == BDFG_i;$ 
5 :    $tmp_{RO} = Count Func OBP(tmp_{BDFG});$ 
6 :   if(RO < tmp_{RO}){
7 :     RO = tmp_{RO};
8 :     RO_{set} = NULL;
9 :     RO_{set} += tmp_{BDFG};
10:   }
11:   elif(RO == tmp_{RO}){
12:     RO_{set} += tmp_{BDFG};
13:   }
14: }
15: }

1 : Maximize Num Dummy calc(RO_{set}, Dummy_{set}){
2 : While(RO_{set} != NULL){
3 :    $tmp_{BDFG} = Select DFG(RO_{set});$ 
4 :   RO_{set} == RO;
5 :    $tmp_{dummy} = Count Dummy calc(tmp_{BDFG});$ 
6 :   if(Dummy < tmp_{dummy}){
7 :     IdleR = tmp_{IdleR};
8 :     Dummy_{set} = NULL;
9 :     Dummy_{set} += tmp_{BDFG};
10:   }
11:   elif(Dummy == tmp_{dummy}){
12:     Dummy_{set} += tmp_{BDFG};
13:   }
14: }
15: }

```

図 2. 全体アルゴリズム

BDFG を抽出し、演算器の観測レジスタ数最大化集合 RO_{set} を作成する(5 行目)。求められた RO_{set} の中から、ダミー演算結果を格納可能なレジスタ数が最も多い BDFG を抽出し、ダミー演算格納可能回数最大化集合 $Dummy_{set}$ を作成する(6 行目)。最後に、得られた $Dummy_{set}$ を返却し、診断容易な特徴を持つ BDFG を求める(8 行目)。

(5 行目)の処理では、各演算器の観測レジスタ数が増加することにより、演算器が多くの観測点でテスト可能となり、タイプ 1 となるハードウェア要素ペアが増加するために行う処理である。図 3 に SUB0_output と MUX2_input0 のハードウェア要素ペア例を示す。回路(a)において SUB0 の観測点集合は {REG1}, MUX2 の観測点集合は {REG1} となり、このハードウェア要素ペアは必ずタイプ 3 となり、識別不能ハードウェア要素ペアとなる。一方、回路(b)においては、SUB0_output の観測点集合は (REG1, REG2), MUX2_input0 の観測点集合は (REG1) となり、異なる観測点集合を持つため、タイプ 1 となり、識別可能ハードウェア要素ペアとなる。

また、(6 行目)の処理では、ダミー演算格納可能回数を最大化することにより、全状態遷移において機能動作でテストしていない動作をテストでき、機能動作時に識別不能ハードウェア要素ペアであったものを識別

可能ハードウェア要素ペアにする。図4に例題回路図および例題回路の機能動作例を示す。図4において機能動作が行われる場合、REG1-REG4及びREG3-REG2の演算ができていない。この演算ができていないことにより、(mux0_1b_0sa, mux1_1b_0sa), (mux0_1b_1sa, mux1_1b_1sa)のペアが識別不能ハードウェア要素ペア(タイプ3)となっている。これらを識別できるようにするには、ある時刻において、SUB0がアイドル状態であり、かつ、その時刻においてアイドル状態のレジスタがあれば、ダミー演算が実行可能である。図4の機能動作から、SUB0は時刻3, 時刻4においてアイドル状態であり、かつ、REG2も時刻3, 時刻4でアイドル状態である。よって、時刻3にREG1-REG4を行い演算結果をREG2に格納、時刻4にREG3-REG2を行い演算結果をREG2に格納するダミー演算として割当てられ、(mux0_1b_0sa, mux1_1b_0sa), (mux0_1b_1sa, mux1_1b_1sa)のペアが識別可能ハードウェア要素ペアとなる。

4. 実験結果

提案手法は、内製高位合成ツールPICTHYのインターフェース機能を用いてC言語で実装し、スキャン設計を仮定した16bitのRTLベンチマーク回路ex2を対象とした。比較対象は、レフトエッジバインディング、文献[7]の手法、提案手法である。なお、提案手法の有効性を評価するために文献[6]の手法の一部を用いて行った。また、全ハードウェア要素ペアのうち、タイプ3として判断されたハードウェア要素ペアの割合も評価を行った。表1に実験結果を示す。

それぞれ、レフトエッジバインディングによって得られた回路がxx-orig、文献[7]の手法を用いて得られた回路がxx-pre、提案手法によって得られた回路がxx-Ansである。表1の結果を見ると、文献[7]の手法はレフトエッジバインディングによって得られた回路よりも識別可能率推定値が高く、さらに提案手法はより高い識別可能率推定値をもつRTL回路を求められている。一方で、識別可能率推定値では、ばらつきが生じているが、得られた解の識別可能率推定値に、ばらつきが生

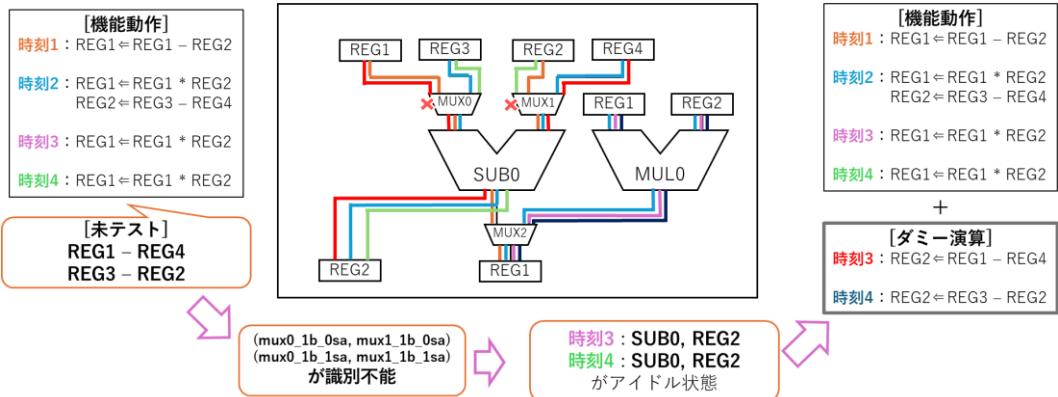
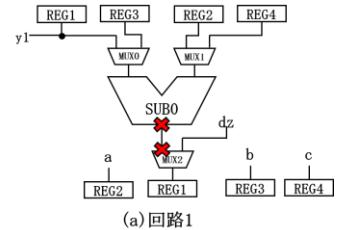
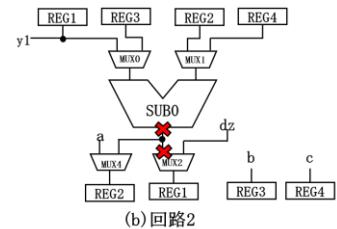


図4. 例題回路図および機能動作例



(a)回路1



(b)回路2

図3. SUB0_output と MUX2_input0 の
ハードウェア要素ペア例

じている。この原因として、提案手法では、ダミー演算格納可能回数の最大化にとどまっており、ダミー演算の割当てを行っていないために生じている。このばらつきは、総故障ペア数が同一であれば、ダミー演算の割当てによって、タイプ3を、タイプ1またはタイプ2へ変化させ、識別可能率推定値を向上させることができる。

5. 考察

本章では、実験結果において、識別可能率推定値にみられるばらつきをダミー演算の割当てによって、タイプ3をタイプ1またはタイプ2へ変化させる方法を考察し、今後の課題を述べる。

総ハードウェア要素ペアのうち、タイプ3であるハードウェア要素ペアをex2-Ans1において、ハードウェアごとに、解析した結果、タイプ3であるいずれのハードウェア(HW)要素ペアでも、一方はMUXの故障であった。このことから、多くのタイプ3である故障ペアはMUXの制御値に適した値を割当てることで、タイプ3からタイプ1またはタイプ2へ変化することができると考え、どのような制御値の割当てを行うことで、変化するのかを解析した。その結果、いくつかのパターンが考えられるため、図5にex2-Ans1回路の一部を示して説明する。

表1. 実験結果

	ダミー演算可能回数	識別可能故障ペア推定数	総故障ペア数	識別可能率推定値 (%)	総HW要素ペア数	タイプ3数	タイプ3割合
ex2-orig	4	109,019,023	109,127,857	99.900270	3403	93	2.732883%
ex2-pre	6	113,065,874	113,172,423	99.905853	3828	109	2.847440%
ex2-Ans1	6	116,592,033	116,701,597	99.906116	4186	128	3.057812%
ex2-Ans2	6	116,592,015	116,701,597	99.906101	4186	127	3.033923%
ex2-Ans3	6	116,592,033	116,701,597	99.906116	4186	128	3.057812%
ex2-Ans4	6	116,592,015	116,701,597	99.906101	4186	127	3.033923%
ex2-Ans5	6	116,592,033	116,701,597	99.906116	4186	128	3.057812%
ex2-Ans6	6	116,592,015	116,701,597	99.906101	4186	127	3.033923%
ex2-Ans7	6	113,065,891	113,172,423	99.905868	3828	110	2.873563%
ex2-Ans8	6	113,065,874	113,172,423	99.905853	3828	109	2.847440%

(パターン 1 : 機能動作時に選択しない演算器の入力

ペアの選択によるタイプの変化)

図 5において、タイプ 3 の HW 要素ペアの一例として(mux10_input0, mux11_input0)がある。これは、機能動作時に(mux10_input0, mux11_input1)のような演算が機能動作で行われないため、タイプ 3 となっている。そのため、制御信号値を(m8, m9, m10, m11)=(X, 1, 0, 1)と割当て、演算結果をレジスタへ格納するダミー演算を行うことで(mux10_input0, mux11_input0)がタイプ 2 へと変化する。

(パターン 2 : 演算器の入力 i を選択した場合、機能動作に入力 i を使用する演算の結果を格納するレジスタと異なるレジスタへ格納するによるタイプの変化)

図 5において、タイプ 3 の HW 要素ペアの一例として(mux11_1b_0sa / mux9_input0)がある。これは、 mux11_input1 を使用する演算において、いずれも mux9_input0 から REG1 へ値を格納しているために、タイプ 3 となっている。そのため、演算に mux11_input1 を使用し、かつ、REG1 以外のレジスタへ格納することが必要であり、制御値を(m8, m9, m10, m11)=(1, X, X, 1)と割当てることで、(mux11_1b_1sa / mux8_input1)はタイプ 1 へ変化する。

(パターン 3 : タイプ 3 から変化なし)

図 5において、タイプ 3 の HW 要素ペアの一例として (mux11_input0 / m11_1sa), (mux11_input1 / m11_0sa)が挙げられる。これらのペアは、どのように制御信号値を割当てたとしても、タイプ 1, タイプ 2 のいずれにも変化させることができない。

このように、タイプ 3 の各 HW 要素ペアに対して、適したダミー演算を割当て、タイプを変化させることで、識別可能率推定値を向上させることができると考えられる。

今後の課題として、最適なダミー演算を挿入するアルゴリズムを追加することで、識別可能率推定値を最大化させることができられる。

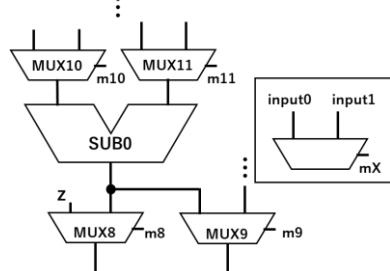


図 5. ex2-Ans1 回路の一部

参考文献

- 1) 藤原 秀雄, ディジタルシステムの設計とテスト, 工学図書株式会社, 東京, 2004.
- 2) 高松雄三, 佐藤康夫, 高橋寛, 桶上喜信, 山崎浩二, “論理回路の故障診断法—外部出力応答に基づく故障箇所指摘法の発展—”, 電子通信学会論文誌 D, vol.J94-D, no.1, pp.266-279, Jan. 2011.
- 3) C. H. Wu, K. j. Lee and S. M. Reddy, “An Efficient Diagnosis-Aware ATPG Procedure to Enhance Diagnosis Resolution and Test Compaction.”, in IEEE Trans. On Very Large Scale Integration (VLSI) Syst., vol.27, no.9, pp.2105-2118, Sept. 2019.
- 4) 青野 竜弥, 細川 利典, 吉村 正義, 山崎 浩二, “完全故障検出効率と完全診断分解能を両立するテストパターン生成法”, 信学技報, vol.124, no.149, DC2024-17, pp.5-10, Jul. 2024.
- 5) 大塚 裕衣, 千田 裕弥, 徐 浩豊, 細川 利典, “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.123, no.146, DC2023-12, pp.25-30, Aug. 2023.
- 6) 大塚 裕衣, 細川 利典, 吉村 正義, 山崎 浩二, “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状態遷移のドントケア割当て手法”, DAシンポジウム 2024 論文集, vol.2024, pp.147-154, Aug. 2024.
- 7) 久保倉 修司, 細川 利典, 山崎 浩二, 吉村正義, “診断容易化レジスタバインディング手法”, 信学技報, vol.124, no.294, DC2025-12, pp.7-12, Dec. 2025.
- 8) Singh, Ravibala, and John Knight. "Concurrent testing in high level synthesis." Proceedings of 7th International Symposium on High-Level Synthesis, IEEE, pp.96-103, 1994.