

疑似ブール最適化を用いた並列テストのためのコントローラの 状態遷移のドントケア割当て法

日大生産工(院) 徐 浩豊 日大生産工(院) 細川 利典
日大生産工(院) 山崎 紘史 日大生産工(院) 新井 雅之
京産大 吉村 正義

1. はじめに

近年、超大規模集積回路(Very Large Scale Integrated Circuits : VLSI)のテストコスト増大に伴い、テストパターン数の削減が重視されている。テストパターン数削減手法にはテスト圧縮法[1-2]やテストパターン数削減のためのテスト容易化設計手法(Design-for-Testability : DFT)[3-6]が提案されてより、多くの故障を並列にテストするテスト並列化によってテストパターン数の削減を行っている。

しかしながら、テスト圧縮法において回路構造が原因となり多くのテストパターン数を削減できない可能性が存在する。また、ゲートレベルでDFT[4-6]を適用すると論理合成後の論理の変更により、遅延の増加や論理合成で実行したタイミングの最適性を損失する可能性がある。以上の理由から、論理合成の適用前の抽象度の高いレジスタ転送レベル(Register Transfer Level : RTL)でテスト並列化を考慮することが重要である。

RTLにおけるテストパターン数削減のためのDFT手法として文献[3]が提案されている。文献[3]では、ハードウェア要素のテスト並列度を高くするための制御信号の生成を無効状態の状態遷移に設計している。そのため、テスト圧縮の効率が向上し、テストパターン数を平均約33%削減できることが報告されている[3]。しかしながら、無効状態の状態遷移の設計によるコントローラ拡大のため、面積オーバーヘッド平均7.11%と大きくなることが報告されている[3]。

本論文では、コントローラの有効状態[3]の状態遷移における制御信号値のドントケアに着目して、データパス中のハードウェア要素のテストの並列度を向上させるためのドントケア割当て法を提案する。各状態遷移の制御信号のドントケアの割当てと各状態遷移で生成するテストパターン数の決定問題を疑似ブール最適化問題(Pseudo- Boolean Optimization : PBO)として定式化する。また、RTL回路における状態遷移で並列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を用いて特定する。

第2章ではRTLデータパスのハードウェア要素の並列テストについて説明し、第3章ではPBOを用いた制御信号のドントケア割当て法を定式化し、説明する。第4章で実験結果を示し、第5章でまとめと今後の課題について述べる。

2. RTLデータパスのハードウェア要素の並列テストに関する

データパスのテストでは、テスト対象ハードウェア要素に対するテスト経路を活性化する必要があり、それを実現するには、コントローラからマルチプレクサ(MUX)やレジスタ(REG)に制御信号を供給する必要がある。テスト経路の活性化処理について本研究では構造的記号シミュレーション[7]を使用する。シミュレーションの結果からデータパスのハードウェア要素の並列テスト可能性をハードウェアテスト可能表でまとめて、ドントケア割当てを実施する。2.1節で構造的記号シミュレーションについて説明する。2.2節でハードウェアテスト可能表について説明する。2.3節でコントローラ制御信号のドントケア割当てについて説明する。

2.1 構造的記号シミュレーション

本節では、データパスにおける状態遷移STで並列テスト可能なハードウェア要素について定義する。

(定義1 : 構造的記号シミュレーション)

以下に述べる手続き1～手続き4までの処理を構造的記号シミュレーションという。

(手続き1) : REGの出力信号線、定数入力の出力信号線、外部出力に接続している信号線に制御可能なシンボル(C シンボル)を割当てる。また、コントローラの状態遷移STの制御信号をデータパスに割当てる。

(手続き2) : 以下の伝搬規則1～3にしたがって、Cシンボルを外部出力方向へ伝搬する。

(伝搬規則1) MUXの制御信号値に対応する入力信号線にCシンボルが割当てられている時、その出力信号線にCシンボルを割当てる。

(伝搬規則2) 演算器のすべての入力信号線にCシンボルが割当てられている時、その出力信号線にCシンボルを割当てる。

(伝搬規則3) 分岐点において、分岐元信号線にCシンボルが割当てられている時、その分岐先信号線にCシンボルを割当てる。

(手続き3) : ホールド機能付きREGがロードモードの時、その入力信号線に観測可能なシンボル(Oシンボル)を割当てる。ホールド機能を持たないREGの入力信号線と外部出力に接続している信号線にOシンボルを割当てる。

(手続き4) : 以下の伝搬規則4～6にしたがって、Oシンボルを外部入力方向へ伝搬する。

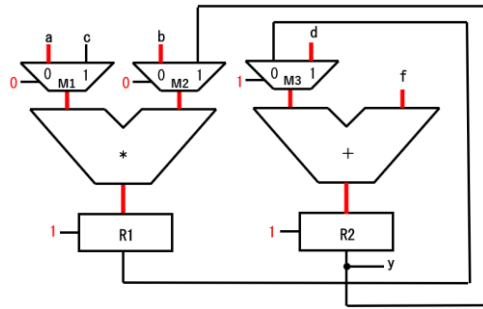


図1. Cシンボル割当て

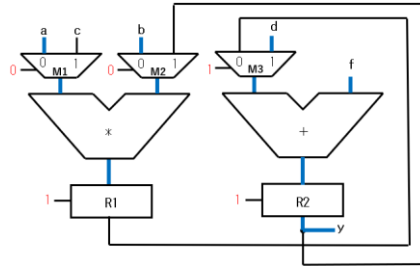


図2. Oシンボル割当て

(伝搬規則4) MUXの出力信号線にOシンボルが割当てられている時、制御信号値に対応する入力信号線にOシンボルを割当てる。

(伝搬規則5) 演算器の出力信号線にOシンボルが割当てられている時、すべての入力信号線にOシンボルを割当てる。

(伝搬規則6) 分岐点において、分岐先信号にOシンボルを割当てられている時、その分岐元信号線にOシンボルを割当てる。

(定義2：状態遷移STで並列テスト可能な演算器H)

状態遷移STで構造的記号シミュレーションを実行した後、データパス中のハードウェア要素Hのすべての入力信号線と出力信号線にCシンボルとOシンボルを割当てられている時、Hは状態遷移STで並列テスト可能であるという。

(定義3：状態遷移STで並列テスト可能な信号線)

状態遷移STで構造的記号シミュレーションを実行した後、CシンボルとOシンボルの両方が割当てられている信号線を状態遷移STで並列テスト可能な信号線という。

(定義4：状態遷移STで並列テスト可能なMUXの制御信号線)

状態遷移STで構造的記号シミュレーションを実行した後、MUXの制御信号値に対応する入力信号線にCシンボルとOシンボルの両方が割当てられ、かつMUXの故障時の制御信号値に対応する入力信号線にCシンボルが割当てられている時、状態遷移STで並列テスト可能なMUXの制御信号線の故障という。

(定義5：状態遷移STで並列テスト可能なREGの制御信号のロードモード故障)

状態遷移STで構造的記号シミュレーションを実行した後、REGの制御信号がホールドモードで、そのREGの入力信号線にCシンボルが割当てられている

表 1. ハードウェアテスト可能表

ADD -in0	ADD -in1	ADD -out	MUL -in0	MUL -in1	MUL -out	M1- 0	M1- 1	M2- 0
○	○	○	○	○	○	○	X	○
M2- 1	M3- 0	M3- 1	m1- s0	m1- s1	m2- s0	m2- s1	m3- s0	m3- s1
X	X	○	X	○	X	○	○	X
R1in	R2in	R1o ut	R2o ut	r1- s0	r1- s1	r2- s0	r2- s1	a
○	○	X	X	○	X	○	X	○
b	c	d	f	y				
○	X	○	X	X				

る時、状態遷移STで並列テスト可能なREGの制御信号のロードモード故障という。

(定義6：状態遷移STで並列テスト可能なREGの制御信号のホールドモード故障)

状態遷移STで構造的記号シミュレーションを実行した後、REGの制御信号がロードモードで、そのREGの入力信号線にCシンボルが割当てられている時、状態遷移STで並列テスト可能なREGの制御信号のホールドモード故障という。

図1は構造的記号シミュレーションの手続き1~2を実行した後の例を示す。図2は手続き3~4を実行した後の例を示す。図1~2において並列テスト可能なハードウェア要素はMUXのM1, M2の0入力信号線と出力信号線, M3の1入力信号線と出力信号線, 乗算器と加算器の入出力信号線, REGのR1, R2の入力信号線である。

2.2 ハードウェアテスト可能表

ハードウェアテスト可能表とは状態遷移STで構造的記号シミュレーションを実行した後、並列テスト可能なハードウェア要素を示した表である。表1は図1のハードウェアテスト可能表の例を示す。表1において「○」は並列テスト可能であることを示す。「×」は並列テスト不能であることを示す。一つの状態遷移で並列テスト可能なハードウェア要素ができる限り多くするように、制御信号値のドントケア(X)に論理値を割当てる説明を2.3節で行う。

2.3 制御信号線のドントケア割当て

コントローラの状態遷移における制御信号にはドントケアが存在する場合がある。レジスタ転送レベルにおけるハードウェア要素の並列テスト可能性を確定させるためには、ドントケア割当てが必要である。

ドントケア割当てとはドントケアに0又は1の論理値を割当てる処理である。ハードウェアテスト可能表はRTL回路のコントローラの中に存在している各状態遷移に対して作成する。図3はコントローラの例である。状態遷移における制御信号はデータパスのハードウェアの制御信号線に出力する。図3に

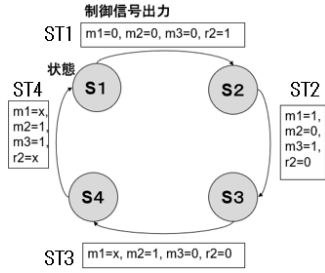


図3. コントローラ

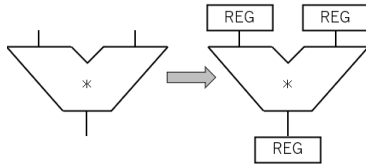


図4. 乗算器の単体テスト

において4つの状態遷移がある．状態遷移 ST3 と ST4 の制御信号値にドントケアが存在することに着目する．ある状態遷移において，制御信号値のドントケア数を d とすると， 2^d のドントケア割当てが存在する．したがって， 2^d 個のハードウェアテスト可能表を作成し，その中の1つを選択する必要がある．図3のコントローラには8個のハードウェアテスト可能表が作成できる．

3. PBOを用いた制御信号のドントケア割当て法

各状態遷移の制御信号のドントケア割当てと各状態遷移で生成するテストパターン数を決定することによって，データパスのテストパターン数を最小化する問題をPBOで定式化する．3.1節は疑似プール最適化問題の説明，3.2節はPBOを用いたドントケア割当ての説明である．

3.1 疑似プール最適化問題

PBO とは Partial Max SAT の一般化で，最適化関数を最小化するように，制約式への論理変数割当てを求める問題である．

各状態遷移のハードウェアテスト可能表を PBO の変数として制約式と最適化関数を定式化することで，各状態遷移のドントケア割当てと各状態遷移で生成するテストパターン数の算出を行い，データパスの見積りテストパターン数最小化問題を解決する．

見積りテストパターンとはデータパスからモジュールを抽出して，単一モジュールのテストを行う環境で生成されたテストパターン数である．図1の乗算器の見積りテストパターン数を計算する時，データパスから乗算器を抽出する．図4のように抽出された乗算器の入出力にホールド機能なしレジスタと

接続する．最後は論理合成してテスト生成を行い，生成されたテストパターン数は乗算器の見積りテストパターン数とする．

3.2 PBOを用いたドントケアとテストパターン数割当て

PBO ソルバの入力は第2章で説明したデータパスのハードウェア要素のテスト可能表の情報である．出力はドントケア割当ての結果と各状態遷移の見積りテストパターン数である．最適化関数は選択された状態遷移の見積りテストパターン数の総和である．

X_{sij} と Y_{si} は導入する変数である． X_{sij} は状態遷移 s の i 番目のドントケア割当てにおいてハードウェア要素 j がテスト可能か否かを判断する変数である．1の時は並列テスト可能(テスト可能表の「○」)を表す，0の時は並列テスト不能(テスト可能表の「×」)である． Y_{si} は状態遷移 s の i 番目のドントケア割当てを選択するか否かを判断する変数である．1の時は選択し，0の時は選択しないことを表す．ドントケアが存在しない状態遷移は常に1になる．

$$\sum_{i=1}^{N_a} Y_{s-i} = 1 \quad (1)$$

式(1)は選択される状態遷移 s のドントケア割当てが1つのみであることを示す． N_a は状態遷移 S のドントケア割当ての場合の数である．

$$\prod_{j=1}^{N_m} \sum_{s=1}^{N_s} \sum_{i=1}^{N_a} Y_{s-i} \times X_{s-i-j} \geq 1 \quad (2)$$

式(2)は全ハードウェア要素が選択した状態遷移で並列テスト可能である式である． N_m はハードウェア要素数， N_s は状態遷移数である．

$$\sum_{s=1}^{N_s} \sum_{i=1}^{N_a} Y_{s-i} \times X_{s-i-j} \times A_s \geq TP_j \quad (3)$$

式(3)はハードウェア要素 j に対して選択された状態遷移のドントケア割当てで生成する必要最小限のテストパターン数の総和の式である． TP_j はハードウェア要素 j を単体テストして，見積りテストパターン数である． A_s は状態遷移 S で生成するテストパターン数である．次に最適化式(4)を説明する．

$$\sum_{s=1}^{N_s} \sum_{i=1}^{N_a} A_s \times Y_{s-i} \quad (4)$$

式(4)は各状態遷移で生成するテストパターン数の総和を最小化する式である．PBOソルバで問題を解く前にもう1つの前処理が必要である．PBOソルバは各変数に対して1,0しか割当てないので， A_s のバイナリーエンコーディングが必要である．式(5)はエンコーディングの式である．前提条件はある状態遷移におけるテスト可能なハードウェア要素は並列にテストされるという前提であるので，各状態遷移で生成されるテストパ

ターン数の上限値は、テスト可能なハードウェア要素の中で最大の TP_j となる。

$$A_s = Z_{s7} \times 128 + Z_{s6} \times 64 + Z_{s5} \times 32 + Z_{s4} \times 16 + Z_{s3} \times 8 + Z_{s2} \times 4 + Z_{s1} \times 2 + Z_{s0} \times 1 \quad (5)$$

制約式(1), (2), (3)最適化式(4), 及びエンコーディング式(5)と変数 X_{sij} , Y_{si} をまとめて, PBOソルバに入力して解を求めると, 各状態遷移が選択するべきドントケア割当てとその状態遷移の生成するテストパターン数が求まり, 必要なテストパターン数が推定される。

4. 実験結果

本章では, ベンチマーク回路で「内製のテスト生成ツール」を使ってテスト生成し, そのテストパターン数を比較する。PBOソルバはClaspである。実験結果を表2, 3に示す。表2はテスト容易化なしの実験結果, 表3は提案手法の実験結果である。対象回路は4つのベンチマーク回路(MAHA, SEHWA, KIM, EX4), 各回路の各状態遷移のドントケア数の上限値を6個に設定し, その他のドントケアにはランダムに論理値を割当てた。提案手法の比較対象としてはドントケア割当てを論理合成に任せて生成した回路のテストパターン数である。1列目は回路名とビット幅, 2列目は全状態遷移の組合せの総数, 内製のテスト生成ツールは等価故障解析した後の代表故障を用いるため, 3列目は総代表故障数, 4列目は検出故障数, 5列目は冗長故障数, 6列目は故障検出率, 7列目は故障検出効率, 8列目は生成されたテストパターン数である。面積オーバーヘッドは平均約1.5%である, KIM(32)以外の回路に対するテストパターン数が削減されたが, KIM(32)回路に対するテストパターン数は逆に9個増加した。原因としてはPBOの解く時間を70000sに設定されたから, 最適解ではない解を使用したことと考える。

大規模なRTL回路のコントローラの状態遷移のドントケア数は膨大になる可能性が高い。ドントケア割当て前に各状態遷移で 2^d 個構造的記号シミュレーションを実行し, ハードウェア要素の並列テスト可能性を評価する必要がある。それゆえ, 全探索には数千万以上の組合せを解く問題になる可能性がある。この場合, 現実的にPBOで解くことは困難であると考え, 一部のドントケアにランダム割当てを導入した。

5. まとめ

本論文では, 疑似ブル最適化を用いた並列テストのためのコントローラの状態遷移のドントケア割当て法を提案した。

今後の課題は, 多くのベンチマーク回路で実験を行うことと, X数が多い回路にはモンテカルロ木探索を導入することが挙げられる。

参考文献

- [1] S.Kajihara, I.Pomeranz, K.Kinoshita: "Cost-Effective Generation of Minimal Test Sets for Stuck-at Faults

in Combinational Logic Circuits,"IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, Vol. 14, Issue12, pp. 1496-1504, Dec.1995.

- [2] S.Kajihara, I.Pomeranz, K.Kinoshita and S.M. Reddy "On Compaction Test Sets by Addition and Removal of Test Vectors, " VLSI Test Symposium, 1994. Proceedings. 12th IEEE, pp. 202-207, Cherry Hill, NJ, The USA, Apr 1994.
- [3] T.Hosokawa, S.Takeda, H.Yamazaki and M.Yoshimura, "A Test Register Assignment Method Based on Controller Augmentation to Reduce the Number of Test Patterns," Proc. Int. Symp. on On-Line Testing and Robust System Design, pp.228-231, 2018.
- [4] M.J.Geuzebroek, J.Th.van der Linden, and A.J.van de Goor, "Test Point Insertion for Compact Test Sets, " Test Conference, 2000. Proceedings. International, pp. 292-301, Atlantic City NJ, The USA, Oct 2000.
- [5] S.Remersaro, J.Rajski, T.Rinderknecht, Sudhakar M.Reddy, I.Pomeranz, "ATPG Heuristics Dependant Observation Point Insertion for Enhanced Compaction and Data Volume Reduction, " IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems, pp. 385-393, Oct. 2008.
- [6] M. Yoshimura, T. Hosokawa, and M. Ohta, "A Test Point Insertion Method to Reduce the Number of Test Patterns, " IEEE the 11th Asian Test Symposium (ATS 2002), pp. 298-304, Nov. 2002.
- [7] 土淵航平, 細川利典, 山崎浩二, "レジスタ転送レベル回路における故障診断容易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, " CPSY2020-62, pp. 73-78, Mar.2021.

表2. テスト容易化なしの実験結果

回路名	組合せ数	総代表故障数	検出故障数	冗長故障数	故障検出率	故障検出効率	テストパターン数
MAHA(32)	1856	7908	7833	75	99.05%	100%	118
SEHWA(32)	1952	9007	8933	74	99.18%	100%	113
KIM(32)	1664	9255	9186	69	99.25%	100%	98
EX4(16)	534	5656	5654	2	99.96%	100%	38
EX4(8)	534	1919	1917	2	99.90%	100%	26

表3. 提案手法の実験結果

回路名	組合せ数	総代表故障数	検出故障数	冗長故障数	故障検出率	故障検出効率	テストパターン数
MAHA(32)	1856	7788	7747	41	99.47%	100%	104
SEHWA(32)	1952	8855	8807	48	99.46%	100%	112
KIM(32)	1664	9315	9273	42	99.55%	100%	107
EX4(16)	534	5707	5705	2	99.96%	100%	36
EX4(8)	534	1961	1959	2	99.90%	100%	25