

ハードウェア並列テスト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 ○ジョ コウホウ 日大生産工(院) 浅見 竜輝
 日大生産工(院) 細川 利典 日大生産工(院) 山崎 紘史
 京産大 吉村 正義 日大生産工(院) 新井 雅之

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]があり、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、ゲートレベルにおけるテスト並列化のためのDFT[3-5]は大規模回路に対して適用する場合、DFT適用箇所の探索に非常に膨大な時間を要する可能性がある、また、ゲートレベルでDFT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。

テストパターン数削減のためのRTLにおけるDFT手法として文献[6]が提案されている。この手法はフルスキャン設計に加えてコントローラ拡大[9]を用いた手法である。

文献[6]ではテストパターン数削減のためのコントローラ拡大を用いたデータパス中のハードウェア要素である演算器とマルチプレクサの並列テストのためのDFT手法が提案されている。しかしながら、文献[6]の手法では、マルチプレクサやレジスタの制御信号線に対する並列テストを考慮していないため、テストパターン数が見積り通りに大きく削減されない回路が存在する。また、並列テストのための専用の状態遷移を無効状態[14]に設計してより、並列テスト使用する状態遷移数が増大すると、状態レジスタを追加し、無効状態数を増加

させている。それゆえ、面積オーバーヘッドが増大する回路が存在する。

本論文では、マルチプレクサやレジスタの制御信号線を含んだデータパス中のすべてのハードウェア要素の並列テストのためのDFT手法を提案する。並列テストに使用する状態遷移は無効状態ではなく、有効状態[8]を使用する。その際、並列テストの効率を向上させるために、有効状態における状態遷移時に出力する制御信号のドントケア割当て法を提案する。

第2章ではハードウェアテスト可能表とコントローラの状態遷移出力のドントケア割当てについて説明し、第3章では疑似ブール最適化問題 (Pseudo-Boolean Optimization: PBO) とPBOを用いた制御信号のドントケア割当て法を説明し、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。

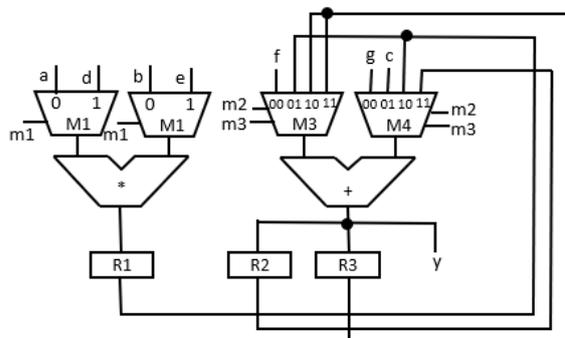


図1. RTL回路のデータパス

加算器	乗算器	M1-0	M1-1	M2-0	M2-1	M3-00	M3-01	M3-10
○	△	△	△	△	△	×	×	○
M3-11	M4-00	M4-01	M4-10	M4-11	R1 in	R2 in	R3 in	M1-s0
×	×	×	○	×	△	×	○	△
M1-s1	M2-s0	M2-s1	M3-s0	M3-s1	R2-s0	R2-s1	a	b
△	○	×	×	○	×	○	△	△
c	d	e	f	g	y	R1 out	R2 out	R3 out
×	△	△	×	×	○	○	×	○

図2. ハードウェアテスト可能表

A Don't Care Filling Method of State Transition Outputs
 on Controllers for Concurrent Hardware Testing

Haofeng XU, Ryuuki ASAMI, Toshinori Hosokawa, Hiroshi YAMAZAKI, Masayoshi YOSHIMURA and Masayuki ARAI

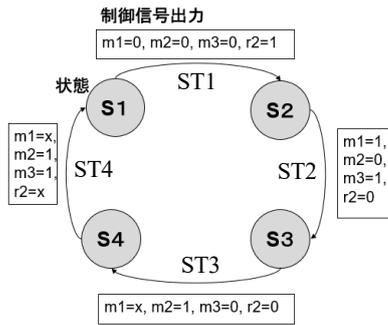


図 3. RTL 回路のコントローラ

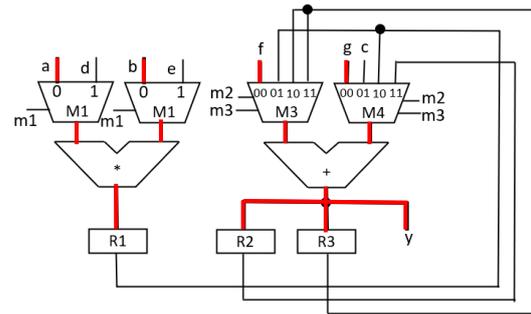


図 4. ST1 状態遷移のデータパス

2. ハードウェアテスト可能表とドントケア割当て

2.1 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テスト可能なハードウェア要素を表した表である。図 1 と図 2 を用いて説明する。図 1 のデータパスは加算器(+)1つ、乗算器(*)1つ、マルチプレクサ(M1, M2, M3, M4)4つとレジスタ(R1, R2, R3)3つで構成される。図 2 のハードウェアテスト可能表の 1, 3, 5, 7 行目は図 1 の RTL データパスのハードウェア要素で、2, 4, 6, 8 行目はテスト可能性に対応する。テスト可能性が「○」はテスト可能を示す。テスト可能性が「△」は制御信号のドントケア割当てたらテスト可能性があることを示す。テスト可能性が「×」がテスト不能を示す。

2.2 制御信号線のドントケア割当て

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状態遷移毎に作成する。図 3 は図 1 のデータパスに対するコントローラである。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 において 4 つの状態遷移があるので、4 つのハードウェアテスト可能表を作成する。状態遷移 ST3 と ST4 の制御信号にはドントケアがあることに着目する。ある状態遷移において制御信号のドントケア数を d とすると、 2^d のドントケア割当てが存在する。ST3 の制御信号はドントケアが ($m1=x$) 1 個存在しているので、($m1=0$) と ($m1=1$) 両方のハードウェアテスト可能表を作成する。ST4 の制御信号はドントケアが ($m1=x, r2=x$) 2 個存在しているので、($m1=0, r2=0$), ($m1=0, r2=1$), ($m1=1, r2=0$), ($m1=1, r2=1$) 4 つのハードウェアテスト可能表を作成する必要がある。

加算器	乗算器	M1-0	M1-1	M2-0	M2-1	M3-00	M3-01	M3-10
○	○	○	×	○	×	○	×	×
M3-11	M4-00	M4-01	M4-10	M4-11	R1 in	R2 in	R3 in	M1-s0
×	○	×	×	×	○	○	○	×
M1-s1	M2-s0	M2-s1	M3-s0	M3-s1	R2-s0	R2-s1	a	b
○	×	○	×	○	○	×	○	○
c	d	e	f	g	y	R1 out	R2 out	R3 out
×	×	×	○	○	○	×	×	×

図 5. ST1 のテスト可能表

3. 擬似ブール最適化問題を用いた制御信号のドントケア割当て法

3.1 擬似ブール最適化問題

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テストパターン数の割当てを行い、テストパターン数最小化問題を解決する。

3.2 PBOを用いたドントケアとテストパターン数割当て

従来のテストレジスタ割当て法[6]はコントローラ有効状態における状態遷移ではなく、無効状態において並列テスト用の状態遷移を追加してテストを行う。それ故に、コントローラの面積が増大する。

本手法はコントローラの有効状態における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の割当てを行って、全ハードウェア要素をテスト可能な最小テストパターン数を出力する。

これから提案手法の手順について説明する。

Step1. コントローラ各状態遷移に対するデータパスのハードウェアテスト可能表を作成する。状態遷移の制御信号にドントケアがある場合は、すべての組合せに対してドントケア割当てを行い、 2^d 個(dはドントケア数)の状態遷移を生成する。 2^d 個の状態遷移に対するハードウェアテスト可能表を作成する。

Step2. 各ハードウェアテスト可能表から、PBOの式を作成する。

Step3. PBOソルバでドントケア割当てされた状態遷移の選択とその選択されたドントケア割当て済み状態遷移で生成するテストパターン数を決定し、テストパターン数最小化問題を解決する。

具体的に各Stepを説明する。Step1は図1のデータパスと図3のコントローラ情報を使う。状態遷移ST1の制御信号(m1=0, m2=0, m3=0, r2=1)をデータパスに与えると、図4に示すように太線の信号線が活性化される。図4から図5のハードウェアテスト可能表が作成できる。

状態遷移ST3の制御信号はドントケアが1個存在するので、ST3-1(x=0)とST3-2(x=1)のハードウェアテスト可能表を追加する。

ST4も同じ操作で4つのハードウェアテスト可能表を追加する。

Step2は変数の導入、制約式と最適化式が必要である。使う情報はドントケア割当て後のハードウェアテスト可能表である。 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の時はテスト可能(テスト可能表の「○」)を表す、0の時はテスト不能(テスト可能表の「×」)である。例で説明すると X_{112} は図5の乗算器に対応している。図5の乗算器は「○」だから $X_{112} = 1$ 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の時は選択する。0の時は選択しない。ドントケアがない状態遷移は常に1になる。ドントケア割当て後ST3からST3-1とST3-2が生成される。 Y_{31} と Y_{32} がそれらを表す。 $Y_{31} = 1$ はST3-1を選択する。制約式は以下の3つがある。

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

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

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

式(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。式(2)は全ハードウェア要素が選択した状態遷移で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。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、必要な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しか割り当てないで、 A_s のバイナリーエンコーディングが必要である。前提条件はハードウェア要素が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。例で説明すると、ある状態遷移Sでテスト可能なハードウェア要素jの最大 TP_j が250とすると、 A_s は8ビットの2進数でエンコーディング可能である。 A_s は以下の式(5)のように表示できる。 $Z_{s0} \sim Z_{s7}$ は0,1変数である。

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

制約式(1), (2), (3)最適化式(4), エンコーディング式(5)と変数 X_{sij} , Y_{si} をまとめて、Step3でPBOソルバに入力して解を求めると、各状態遷移が選択すべきドントケア割当てとその状態遷移で生成するテストパターン数が求まる。

表1. 実験結果

method	Circuit	All	Detect	Redundant	TP
None	Ex-2	21311	21274	37	65
Proposed	Ex-2	21311	21271	42	77

4. 実験結果

本章では、ex2回路に対して、「ドントケア割当を論理合成にする」と「提案手法のドントケア割当を行った」、この2つ回路で制約条件なしでテスト生成を行い、総故障数、検出できる故障数と必要なテストパターン数を評価した。実験結果を表1に示す。表1の1行目は論理合成でドントケア割当てをする結果である。2行目は提案手法でドントケア割当てをする結果である。3列目は回路の総故障数、4列目は検出故障数、5列目は冗長故障数、6列目はテスト生成したテストパターン数である。提案手法のテストパターン数は論理合成より高いけど、生成したテストパターンを解析して、無効状態の状態遷移でテストパターン16個を生成している。有効状態の状態遷移は61個のテストパターンを生成している。これは、一般的なテスト生成ではなくて、RTLでのDFTの情報を考慮した専用のテスト生成が必要であることを考える。

5. まとめ

本論文では、コントローラにおけるドントケア割当てに基づくテスト容易化設計を説明した。本手法では文献[6]で考慮していないマルチプレクサやレジスタの制御信号線を含んだデータパス中のすべてのハードウェア要素の並列テストのためのDFT手法である。

今後の課題は、多くの回路で実験を行うことと、制御信号のドントケア数が多い回路への対応が挙げられる。

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] 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.

[4] 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.

[5] 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.

[6] 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.

[7] Kedarnath J. Balakrishnam and Lei Fang, "RTL Test Point Insertion to Reduce Delay Test Volume, " 25th IEEE VLSI Test Symposium (VTS' 07) , pp. 325-332 , May. 2007.

[8] S. Ohtake, T. Masuzawa, and H. Fujiwara, " A non-scan approach to DFT for Controllers Achieving 100% Fault Efficiency, " Journal of Electronic Testing: Theory and Applications (JETTA), Vol. 16, No. 5, pp. 553-556, Oct. 2000.

[9] L.M. FLottes, B. Rouzeyre, L. Volpe, "A Controller resynthesis based method for improving datapath testability, " IEEE International Symposium on Circuits and Systems, pp. 347-350, May 2000.