

システムLSI時代における新テスト技術

杉原, 真
九州大学大学院システム情報科学研究科情報工学専攻

安浦, 寛人
九州大学大学院システム情報科学研究科情報工学専攻

<https://hdl.handle.net/2324/7654>

出版情報：情報処理学会論文誌. 42 (3), pp.409-418, 2001-03-15. 情報処理学会
バージョン：
権利関係：

システム LSI 時代における新テスト技術

杉原 真[†] 安浦 寛人[†]

本論文では、システム LSI 時代の要素技術となる新たなテスト手法である CBET テスト手法について議論する。CBET は BIST と ATPG を併せて用いるものであり、テスト時間を大幅に削減する。CBET は外部入出力ピン数の制限を緩和し、テスト時に必要とされる LSI テスタのメモリサイズも削減する。CBET テスト手法の有効性を理論的かつ実験的に示す。さらに、システム LSI を設計する上で主流となるであろうコア・ベース設計に CBET を適用する。コア・ベース LSI のテスト時間最小化問題を定義し、効率の良いアルゴリズムを与える。実験において、既存のテスト手法によるコア・ベース LSI のテスト時間を提案手法は最高 97%以上削減した。また、実験によって得た計算時間は本稿のアルゴリズムが効率の良いものであることを示している。

A Novel Test Technique in the SOC Era

MAKOTO SUGIHARA and HIROTO YASUURA

In this paper, CBET test approach, a novel test methodology which is a key technology in the SOC era is discussed. CBET test approach adopts both BIST and ATPG and can reduce much test time. It can also alleviate the restriction of external pin count and reduce memory size on LSI tester. A validity of the method is shown theoretically and experimentally. CBET is extended to core-based design method which will be a major design method in designing SOC. A test time minimization problem is defined and an efficient algorithm for the problem is given. The method reduces up to 90 % of test time by traditional test method in experiments. And calculation time for our algorithm validates its effectiveness.

1. はじめに

近年の目覚ましい LSI 技術の進歩によって 1 チップ上に搭載されるトランジスタ数は飛躍的に増加している。大規模なシステムを 1 チップ上に実現するシステム LSI が注目を浴びている。システム LSI の設計および検証に要する時間を節約するために、多くのあらかじめ設計と検証がなされたコアと呼ばれる回路ブロックが用いられる。いわゆるコア・ベース設計である。

トランジスタ数の増加によって適用すべきテストベクタ数は増加する。テストベクタ数の増加はテスト時間の増加を引き起こし、それゆえに、チップあたりのテストコストを高める。このために、テスト時間を削減しようとする多くの研究が行われてきた。テスト生成の分野での研究者の主な関心はテスト集合圧縮技術で

ある [5, 11, 15]。これらの研究の目標は可能な限り小さなテスト集合を達成することである。組み込み自己テスト (以下、BIST ...) の分野では、random-pattern-resistant faults [12] を容易に検出する技術が研究されてきた。これは、BIST 回路の生成するテストベクタをある程度決定的にしようというものである。最近ではテスト時間を削減するために、テストスケジューリングに関する研究がさかんに行われている。Aertsらはスキャンテストのための一テストスケジューリング手法を提案している [14]。この手法は BIST を用いない場合に非常に有効なテスト時間削減が行えるが、チップ内部とチップ外部との間の速度差が著しい LSI に対しては十分なものではない。[8-10, 13] では BIST と外部テストを併せて用いるテスト手法 (CBET ...) におけるテストスケジューリング手法が提案されている。CBET に基づく複数のテスト集合が各コアに対して生成されており、設計する LSI のテスト時間を最小にするように、各コアの最適なテスト集合が選択

[†] 九州大学大学院システム情報科学研究科情報工学専攻
Department of Computer Science and Communication
Engineering Graduate School of Information Science
and Electrical Engineering, Kyushu University
英語圏では SOC(System-On-a-Chip) と呼ばれる。

Built-In Self Test の頭字語
Combination of BIST and External Test の頭字語。

される。これらの手法は [14] の手法に較べて、必要とする外部入出力ピン数を抑えることができ、テスト時間も多に削減する。しかしながら、BIST に要する面積において不利となる。本稿では [8-10] をまとめるとともに、CBET テスト手法の今後の課題について述べる。

本稿では CBET テスト手法について議論する。2 節で CBET テスト手法によるテスト時間の削減の概観を与える [8]。3 節では CBET テスト手法によるテスト時間を理論的かつ実験的に与える [8]。本解析は CBET テスト手法のいくつかの特性を明らかにする。さらに、4 節でコア・ベース LSI のテスト時間最小化問題を定義し [9, 10]、5 節で効率の良いテスト時間最小化アルゴリズムについて議論する [8]。6 節でコア・ベース LSI を CBET によって行うことの有効性を示す。7 節で結論と今後の課題を述べる。

2. CBET テスト

CBET は Combination of BIST and External Test の頭字語である。CBET テスト手法とは狭義には BIST と外部テストを組合せて用いる方法と定義され、広義には簡単な回路による BIST と ATPG によるテストベクタを組合せて用いる手法と定義される。CBET テスト手法はテスト時間を大幅に削減するだけでなく、LSI テスタ上のピンメモリサイズも小さなものとする。CBET テスト手法は BIST と外部テストの回路構造を制限するものではない。すなわち、BIST の回路に LFSR やカウンタを用いても良いし、外部テストにスキャンチェーンやテストバスを用いても良い。また、ATPG による外部テストのテストベクタはチップの ROM に置かれても良い。ROM に ATPG のテストベクタを置くような方法は一種の BIST とも言えるが、それでもなお、通常の BIST とはその伝搬方法に一線を画するものがあるために、CBET テストにおいては外部テストに分類される。

テストベクタを適用する速度はテスト時間にどのような影響を及ぼすのであろうか？ 本稿では、一つのテストベクタはテストするために適用される被テスト回路 (以下 CUT) の主入出力の値と定義する。BIST と外部テストの構造は図 1 のように示される [6]。破線で描かれた四角は外部テストにおけるチップと LSI テ

スタの境界を示し、実線で描かれた四角は BIST でのそれを示す。両テスト間でこれらの境界に起因する差異が大きく二つ存在する。第一の差異はテストベクタが置かれる場所である。BIST のテストベクタはチップ内に置かれ、外部テストのテストベクタはチップ外に置かれる。テストベクタの置かれる場所の差異は両テスト間でクロック周波数の差異を生じる可能性がある。もちろん、外部テストのテストベクタがチップ上に置かれるならば、すなわち、ATPG によるテストベクタが ROM 化されるのならば、両テスト間でクロック周波数の違いはないかもしれない。もう一つの差異は一テストベクタを伝搬するために必要なクロックサイクル数である。BIST のテストベクタは外部テストのテストベクタより CUT のそばに置くことができる。それゆえに、BIST のテストベクタは外部テストよりも少ないクロックサイクル数で適用できる。

BIST と外部テストのそれぞれのテストベクタの質はテスト時間にどのような影響を及ぼすのであろうか？ ここで、故障検出率 c を検出すべき故障数に対する検出済の故障数の割合とし、故障検出率 c での一テストベクタ当りのテストの質 $Q(c)$ を故障検出率 c で適用されるテストベクタによって検出される故障数とする。以下、両テスト手法のテストの質について考える。テストベクタのテストの質を非常に高める多くの良い技術が存在するために ATPG はテストの質において最高度のテストベクタを生成すると考えることができる。一方、BIST は CUT への入力は疑似乱数に基づくものであるために、テストの質が非常に悪いテストベクタを生成する。16 ビット乗算器において、実験的に

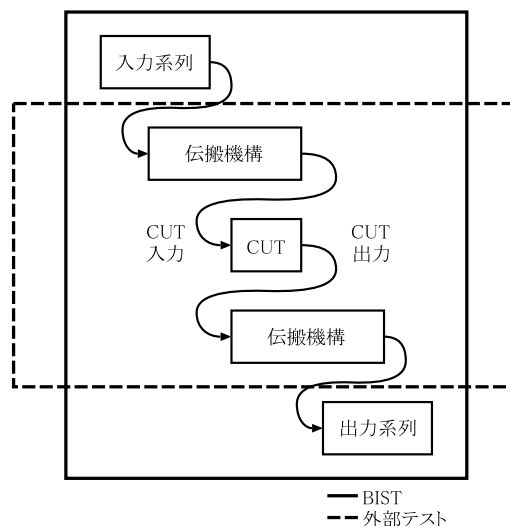


図 1 テストの構造

Automatic Test Pattern Generation の頭字語。通常、外部テストに用いられるテストパターンは ATPG により得る。
Linear Feedback Shift Register の頭字語。線形帰還シフトレジスタ。
Circuit Under Test の頭字語、被テスト回路。

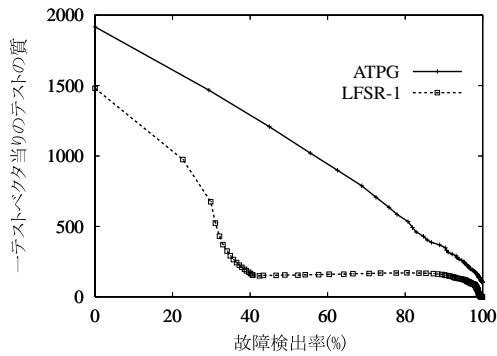


図2 故障検出率に対するテストの質 (0-100%)

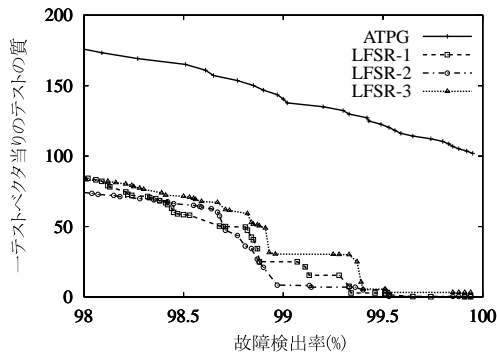


図3 故障検出率に対するテストの質 (98-100%)

得られる両テスト手法のテストの質を図2と図3に示す。これらの図は故障検出率に対するテストの質の平均値を示す。図中のATPGは外部テストのテストの質を示す。LFSR-1, LFSR-2, および, LFSR-3は互いに原始多項式の異なるLFSRであり, 初期値は全一である。図3は図2の故障検出率が高い部分を拡大したものである。横軸は故障検出率を示し, 縦軸は一テストベクタあたりのテストの質の平均値を示す。ある故障検出率でのテストの質の平均値は三つの値を用いて計算した。その故障検出率を達成するために適用されたテストベクタ数, さらに故障を検出するために適用されるテストベクタ数, および, 新たに適用されたテストベクタによって検出された故障数である。図2は外部テストのテストの質 $Q_E(c)$ がBISTのテストの質 $Q_B(c)$ より高いことを示している。図3は高い故障検出率において, ATPGによるテストベクタはある程度のテストの質を保持することができるが, BISTは保持することができないことを示している。高い故障検出率において, BISTは故障を検出しない膨大な量のテストベクタを適用することになる。いずれのテスト手法も99.97%の故障検出率を達成しているが, 外部テストとBISTの必要とされるテストベクタ数はそれぞれ64および39667となり, 同程度の

故障検出率を達成するためにBISTは外部テストよりも多くのテストベクタを必要とすることがわかる。

以上で議論したように, BISTはクロック周波数や一テストベクタあたりのクロックサイクル数で有利であり, 外部テストはテストの質で有利である。BISTのクロック周波数 F_B は外部テストのクロック周波数 F_E 以上であり, BISTで一テストベクタあたりのクロックサイクル数 AC_B は外部テストの一テストベクタあたりのクロックサイクル数 AC_E よりも小さい。テストの速さを単位時間に適用されるテストベクタ数と定義し, $S = F/AC$ で与える。ここで, S_B と S_E をそれぞれBISTと外部テストのテストの速さとする。もし, $S_B = S_E$ であり, かつ, どのような故障検出率 c においても $Q_E(c) > Q_B(c)$ が成り立つのならば, テストをするために外部テストを用いれば良い。もし, $S_B \gg S_E$ ならば, 外部テストを使う必要はないかもしれない。直観的に言って, もし, $Q_B(c) \cdot S_B = Q_E(c) \cdot S_E$ ならば, BISTは故障検出率 c を達成するまで適用されるべきであり, その後は所望の故障検出率を達成するまで外部テストを適用すれば良い。CBETテスト手法ではテストの過程に応じて外部テストとBISTを切り替えて適用される。

3. CBETによるテスト時間削減の解析

本節では, CBETテスト手法によるテスト時間の削減を理論的かつ実験的に解析する。

3.1 仮定

CBETテスト手法によるテスト時間削減を解析する上で, 以下の仮定を用いる。

- 冗長故障は考慮しない。
- BISTにLFSRを用いる。LFSRの特性多項式は原始多項式である。すなわち, LFSRはM系列を発生する。
- LFSRのビット幅 l はCUTの入力数に等しい。
- 全零の入力によってのみ検出される故障は存在しない。
- ATPGによるテストベクタ数は最大独立故障集合の要素数に等しい。
- BISTの機構において一テストベクタ当りのクロックサイクル数 AC_B は定数である。
- 外部テストの機構において一テストベクタ当りのクロックサイクル数 AC_E は定数である。
- BISTのテストベクタは1クロックサイクルで適用される。すなわち, $AC_B = 1$ である。

3.2 記法

CBETテスト手法によるテスト時間削減を解析す

るために以下の記法を用いる．

- \mathbf{F} : 全ての故障からなる集合．
 $\mathbf{F} = \{f_1, f_2, \dots, f_{|\mathbf{F}|}\}$
- \mathbf{TV}_f : 故障 f を検出できる全てのテストベクタからなる集合．
- tv_f : 集合 \mathbf{TV}_f に含まれるあるテストベクタ．
- \mathbf{IF} : 独立故障集合
 $\mathbf{IF} = \{if_1, if_2, \dots, if_{|\mathbf{IF}|}\}$
- $F(v)$: テストベクタ v によって検出される故障の集合．
- $P(A_i)$: 事象 A に関する A_i の生起確率．
- D_x^f : x サイクル以内で故障 f が検出されるという事象．
- N_x^f : x サイクル以内で故障 f が検出されないという事象．
- $P_{A_i}(B_j)$: A に関する事象が A_i であるとき, B に関する事象が B_j であるという条件付き確率．

3.3 理論的解析

本節では BIST, 外部テスト, および, CBET のテストベクタ数に対する故障検出率を解析する．

まず, BIST でのテストベクタ数に対する故障検出率について解析する．BIST によって故障 f を検出するとき, 故障 f が x サイクル以内で検出されない確率は以下の式で表される．

$$P(N_x^f) = P(N_1^f) \cdot P_{N_1}(N_2^f) \cdot \dots \cdot P_{N_{x-1}}(N_x^f) \\ = \prod_{i=1}^x \frac{2^i - 1 - (i-1) - |\mathbf{TV}_f|}{2^i - 1 - (i-1)} \quad (1)$$

式 (1) より, 故障 f が x サイクル以内で検出される確率は以下の式で表される．

$$P(D_x^f) = 1 - P(N_x^f) \\ = 1 - \prod_{i=1}^x \frac{2^i - 1 - (i-1) - |\mathbf{TV}_f|}{2^i - 1 - (i-1)} \quad (2)$$

式 (2) より, x サイクルの BIST を行うことによって得られる故障検出率は以下の式で表される．

$$FC_B(x) = \frac{1}{|\mathbf{F}|} \sum_{\forall f \in \mathbf{F}} P(D_x^f) \quad (3)$$

当然ながら, 式 (3) は潜在的にモデル化誤差を含む．現実では各故障の検出は 0 か 1 で表されるべきであるが, 本稿のモデルでは各故障の検出を 0 から 1 までの連続値で表している．もし, 故障 f が x 個以内の入力で検出されるのならば, 故障検出率の誤差は $1 - P(D_x^f)$ である．もし, 故障 f が x 個以内の入力で検出されないのであれば, 故障検出率の誤差は $P(D_x^f)$ である．これらの誤差の確率はそれぞれ $P(D_x^f)$, お

よび, $1 - P(D_x^f)$ である．確率の観点から式 (3) は

$$\pm \sum_{\forall f \in \mathbf{F}} P(D_x^f) \{1 - P(D_x^f)\}. \quad (4)$$

の誤差を含むことになる．

図 4 は ISCAS'85 ベンチマーク回路の C1355 におけるクロックサイクル数に対する故障検出率を示す．実線は式 (3) から計算される故障検出率の理論値を示し, 破線は様々な LFSR によって達成される故障検出率を示す．ここで, LFSR の初期値は全てのビットが 1 である．式 (3) の誤差の範囲は BIST の初期では大きいものであるが, BIST を行えば行うほど小さなものとなる．

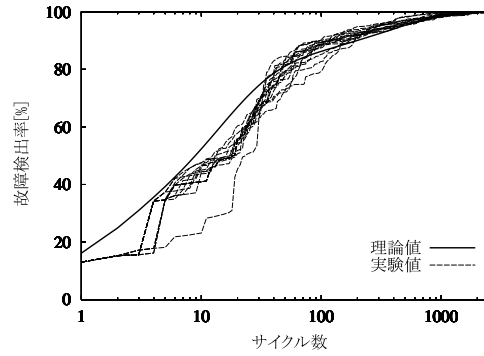


図 4 BIST におけるクロックサイクル数に対する故障検出率

式 (2) より, BIST の一テストベクタあたりのテストの質 $Q_B(FC_B(x))$ は以下ようになる．

$$Q_B(FC_B(x)) = \sum_{\forall f \in \mathbf{F}} \{P(D_{x+1}^f) - P(D_x^f)\} \quad (5)$$

次に, 外部テストにおけるテストベクタに対する故障検出率について解析する．本解析では, 3.1 節のように故障の独立性の概念が用いられる．すなわち, 外部テストのテストベクタ数は最大独立故障集合の要素数に等しいと仮定される．故障検出率 0% のときにテストベクタ tv_{if} が回路に適用されると, 故障検出率の増分は $|F(tv_{if})|/|\mathbf{F}|$ となる．回路にテストベクタが $tv_{if_1}, tv_{if_2}, \dots, tv_{if_n}$ といった順で適用されるとき, x サイクルで達成される故障検出率 $FC_E(x)$ は以下の式で表される．

$$FC_E(x) = \begin{cases} 0 & (0 \leq x < AC_E) \\ \frac{1}{|\mathbf{F}|} \left| \bigcup_{i=1}^{\lfloor x/AC_E \rfloor} F(tv_{if_i}) \right| & (1 \leq \frac{x}{AC_E} \leq |\mathbf{IF}|) \end{cases} \quad (6)$$

最後に CBET テスト手法でのテスト時間について解析する．解析を容易にするために, 外部テストは BIST のあとで行われると仮定する．独立故障 if が x サイクル以内の BIST で検出される確率 $P(D_x^{if})$ は以下の

式で表される．

$$P(D_x^{if}) = 1 - P(N_x^{if}) \\ = 1 - \prod_{i=1}^x \frac{2^i - 1 - (i-1) - |\mathbf{TV}_{if}|}{2^i - 1 - (i-1)} \quad (7)$$

上式より，残りの独立故障数 $R(x)$ は以下のようになる．

$$R(x) = |\mathbf{IF}| - \sum_{\forall if \in \mathbf{IF}} P(D_x^{if}) \quad (8)$$

式 (7) より， $|\mathbf{TV}_{if}|$ が大きければ大きいほど独立故障 if は検出されやすいことが理解される．よって，BIST を x サイクルだけ行ったあとに残る独立故障は独立故障集合 \mathbf{IF} のうちで最も検出が困難な $R(x)$ 個の独立故障であると言える．ここで， $i < j$ ならば， $|\mathbf{TV}_{if_i}| < |\mathbf{TV}_{if_j}|$ であると仮定する． x サイクルの BIST のあとの外部テストのテストの質は以下のように表される．

$$Q_E(FC_B(x)) = \left| F(tv_{if_{R(x)}}) - \bigcup_{i=R(x)+1}^{|\mathbf{IF}|} F(tv_{if_i}) \right| \quad (9)$$

x サイクルの BIST を行う CBET テスト手法によって達成されるテスト時間 $T(x)$ は以下のようになる．

$$T(x) = \frac{x}{S_B} + \frac{R(x)}{S_E} \quad (10)$$

もし，

$$Q_E(x) \cdot S_E > Q_B(x) \cdot S_B, \quad (11)$$

を満たす BIST のサイクル数 x が存在すれば，CBET テスト手法によってテスト時間を削減することができる．本節の解析は容易に外部テストのテストベクタが ROM にある設計に拡張できる．すなわち， F_B と F_E を同様に取り扱いえば良い．

3.4 実験的解析

本節では CBET テスト手法によるテスト時間を実験的に解析する．実験に用いた回路の特性は表 1 に示される．実験に用いた回路は ISCAS'85 ベンチマーク回路のものである．図 5 は 100% の故障検出率を

表 1 回路の諸元

回路名	入力線数	出力線数	セル数	故障数
C432	36	7	157	513
C499	41	32	202	750
C880	60	26	383	942
C1355	41	32	546	1566
C1908	33	25	878	1862
C3540	50	22	1620	3126

達成する CBET に基づいたテスト集合の外部テストと BIST のテストベクタ数の関係を表すものである．

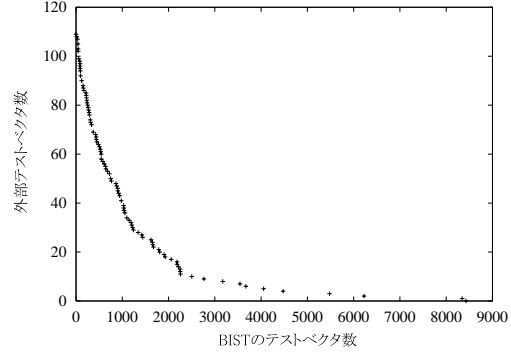


図 5 C1908 における CBET に基づくテスト集合

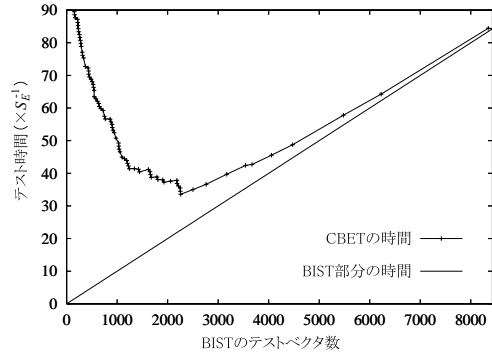


図 6 C1908 における CBET によるテスト時間

横軸は BIST のテストベクタ数で，縦軸は BIST のテストベクタ数である．BIST のサイクル数を多くしてもそのテストの質が悪く，そのようなときは外部テストのテストの質が BIST と比べて非常に良いことが理解される．図 6 は $S_B/S_E = 100$ であるときの CBET に基づく様々なテスト集合によって成し遂げられる C1908 のテスト時間を示す．たとえば， $S_B = S_E$ であるとしても，BIST と比べて外部テストは一テストベクタを適用するのに多くのクロック数を要するため，上記のテストの速さの比の値は妥当なものであると言える．横軸はテスト集合に含まれる BIST のテストベクタ数であり，縦軸はテスト時間である．最小のテスト時間は $33.6 \times S_E^{-1}$ である．CBET テスト手法は BIST だけでテストを行うときのテスト時間を 60.17% だけ削減した．また，外部テストだけでテストを行うときのテスト時間を 70.0% だけ削減した．2 節で述べたように，速度比 S_B/S_E がある範囲内にあるときにだけ，CBET によってテスト時間の削減ができる．実験的に得られるその上限値を表 3.4 に示す．

4. テスト時間最小化問題

本節では，コアベース LSI のテスト時間最小化問題

表 2 速度比の上限値

	C432	C499	C880	
S_B/S_E	455.0	261.0	567.4	
		C1355	C1908	C3540
		394.0	1101.5	578.9

を定義する [9,10] . 本問題を解くことはコア・ベース LSI のテスト時間を削減する上で非常に有効である .

4.1 仮定

テスト時間最小化問題を定義する上で、ターゲットとするシステム LSI は以下の仮定を満たす .

- コア・ベース LSI は n 個のコアから構成される .
- 各コアに対して、十分な故障検出率を達成するテスト集合が複数個与えられる .
- 各コアに対して CBET に基づくテスト集合が一個以上与えられる . 同じコアに対して与えられる複数個のテスト集合は互いに BIST と外部テストのテストパターン数の点で異なる .
- BIST の動作周波数は f_B であり、外部テストの動作周波数は f_E である . この二つの動作周波数は $f_B \geq f_E$ の関係を持つ .
- 各コアのテスト手法を切替える上でオーバーヘッドは存在しない .
- 外部入出力は同時に二つ以上のコアによって占有されない . 外部入出力は各コア間で時分割に用いられる .

本手法はスキャン設計やテストバス方式に容易に拡張することができる . さらに、本手法はアイソレーション・リング [2,4] , 透過法 [3] , および、グリッド・ベースの直接アクセス手法 [1] にわずかな拡張で適用可能である . 電力消費が許容できない場合にも解を求める上でその制約を課すだけで対処できる . このような制約は我々の基本的発想を明瞭に説明するために仮定しない .

4.2 記法

テスト時間最小化問題を定義する上で以下の仮定を設ける .

- V_i : コア i の CBET に基づく複数個のテスト集合を要素とする集合 .
- v_i : コア i のテスト集合 .
- v : 全てのコアに選択されたテスト集合を表すベクトル . $v = (v_1, v_2, \dots, v_n)$
- F_B : BIST の動作周波数 .
- F_E : 外部テストの動作周波数 .
- $C_E(v_i)$: テスト集合 v_i に含まれる外部テスト部分の適用に要するクロックサイクル数 .

- $C_B(v_i)$: テスト集合 v_i に含まれる BIST 部分の適用に要するクロックサイクル数 .
- $C_C(v_i)$: テスト集合 v_i を適用するのに要するクロックサイクル数 . $C_C(v_i) = C_E(v_i) + C_B(v_i)$.
- $T_E(v_i)$: コア i のテスト集合 v_i に含まれる外部テストのテストベクタを適用するのに要する時間 .
- $T_B(v_i)$: コア i のテスト集合 v_i に含まれる BIST のテストベクタを適用するのに要する時間 .
- $T_C(v_i)$: コア i のテスト集合 v_i に含まれる全てのテストベクタを適用するのに要する時間 .
- $T_{SUM}(v)$: 各コアに選択されたテスト集合 $v_i (1 \leq i \leq n)$ の外部テストに要する時間の和 .
- $T_{MAX}(v)$: 各コアに選択されたテスト集合 $v_i (1 \leq i \leq n)$ のなかで最も適用に要するテスト集合の適用時間 .
- $T(v)$: 各コアにテスト集合 $v_i (1 \leq i \leq n)$ が与えられたときの LSI のテスト時間 .

4.3 問題定義

テスト時間最小化問題とはコア i に複数個のテスト集合を要素とする集合 V_i が与えられたときに、テスト時間 $T(v)$ を最小化するテスト集合ベクトル v を求める問題と定義される .

図 7 に示されるように、全てのコアの外部テストの時間の和 $T_{SUM}(v)$, あるいは、 $v_i (1 \leq i \leq n)$ のなかでその適用時間が最も大きいものの時間 $T_{MAX}(v)$ が LSI のテスト時間 $T(v)$ となる . 全てのコアの外部テストの時間の和 $T_{SUM}(v)$ は以下の式で表される .

$$T_{SUM}(v) = \sum_{i=1}^n T_E(v_i) = \sum_{i=1}^n \frac{C_E(v_i)}{F_E} \quad (12)$$

コア i のテスト集合 v_i の適用に要する時間 $T_C(v_i)$ は以下の式で表される .

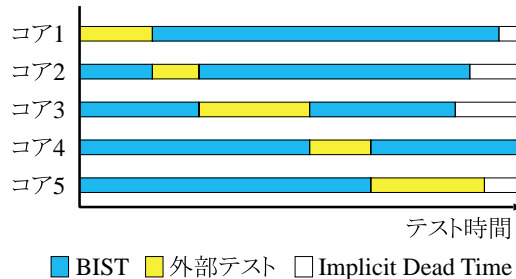


図 7 テスト時間最小化のスケジューリング例

$$T_C(v_i) = T_E(v_i) + T_B(v_i) \\ = \frac{C_E(v_i)}{F_E} + \frac{C_B(v_i)}{F_B} \quad (13)$$

式 (13) より, $v_i (1 \leq i \leq n)$ のなかでその適用に最も時間を要するテスト集合の時間 T_{MAX} は以下の式で表される.

$$T_{MAX}(v) = \max_{i=1}^n T_C(v_i) \\ = \max_{i=1}^n \left\{ \frac{C_E(v_i)}{F_E} + \frac{C_B(v_i)}{F_B} \right\} \quad (14)$$

式 (12), および (14) から目的関数 $T(v)$ は以下のよう表される.

$$T(v) = \max \{ T_{SUM}(v), T_{MAX}(v) \} \\ = \max \left[\sum_{i=1}^n \frac{C_E(v_i)}{F_E}, \max_{i=1}^n \left\{ \frac{C_E(v_i)}{F_E} + \frac{C_B(v_i)}{F_B} \right\} \right] \quad (15)$$

テスト時間最小化問題は目的関数 $T(v)$ を最小化する $v = (v_1, v_2, \dots, v_n)$ を探索することによって解かれる. これは典型的な組合わせ最適化問題である.

5. テスト時間最小化アルゴリズム

本節では 4 節の問題を解く効率の良い近似アルゴリズムを与える.

5.1 記法

本節では 4.2 節で用いられた記法を用いる. また, 効率の良いアルゴリズムを設計するために以下の記法を併せて用いる.

- $\mathbf{BT}_i = \{T_B(v_i) \mid \forall v_i \in \mathbf{V}_i\}$
- $bt_i \in \mathbf{BT}_i$
- $ET(bt_i)$: BIST に要するテスト時間が bt_i であるテスト集合の外部テストに要する時間.
- $ST(bt_i) = bt_i + ET(bt_i)$
- $\mathbf{M}_i = \left\{ x \mid \forall x \in \mathbf{V}_i, T_C(x) = \min_{v_i \in \mathbf{V}_i} T_C(v_i) \right\}$
- $m_i \in \mathbf{M}_i$

5.2 仮定

組合わせ最適化問題を効率よく解くためには枝刈りが非常に重要である. 枝刈りの目的で, 以下の仮定を設ける:

- $|\mathbf{M}_i| = 1$
- $0 \leq bt_i \leq T_C(m_i)$ において, $ST(bt_i)$ は単調減少する.
- $T_C(m_i) \leq bt_i$ において, $ST(bt_i)$ は単調増加する.
- $ET(bt_i)$ は単調減少する.

5.3 定理

補題 1 最小化されるテスト時間は $\max_{i=1}^n T_C(m_i)$ 以上である.

証明 三番目の仮定より, 以下の式が導かれる.

$$T(v) = \max \left\{ \sum_{i=1}^n T_E(v_i), \max_{i=1}^n T_C(v_i) \right\} \\ \geq \max_{i=1}^n T_C(v_i) \\ \geq \max_{i=1}^n T_C(m_i)$$

□

定理 1 もし, $\exists v_i \in \{v \mid v \in V_i, T_B(v) < T_B(m_i)\}$ がテスト時間最小化問題の解を与えるのならば, v_i と m_i を交換することができる.

証明 今テスト集合ベクトル $v = (v_1, \dots, v_i, \dots, v_n)$ が $T(v)$ を最小化するとしよう. 二番目と四番目の仮定より, 以下の式が導かれる.

$$T_C(\forall x \in \{v \mid v \in V_i, T_B(v) < T_B(m_i)\}) > T_S(m_i) \quad (16)$$

$$T_E(\forall x \in \{v \mid v \in V_i, T_B(v) < T_B(m_i)\}) > T_E(m_i) \quad (17)$$

式 (16) と四番目の仮定より, 以下の式が得られる.

$$\sum_{i=1}^n T_E(v_i) = T_E(v_1) + \dots + T_E(v_i) + \dots + T_E(v_n) \\ \leq T_E(v_1) + \dots + T_E(m_i) + \dots + T_E(v_n) \quad (18)$$

式 (17) と二番目の仮定より, 以下の式が得られる.

$$\max_{i=1}^n T_C(v_i) = \max \{T_C(v_1), \dots, T_C(v_i), \dots, T_C(v_n)\} \\ \geq \max \{T_C(v_1), \dots, T_C(m_i), \dots, T_C(v_n)\} \quad (19)$$

式 (18) と (19) より, 以下の式が得られる.

$$T(v) = \max \left\{ \sum_{i=1}^n T_E(v_i), \max_{i=1}^n T_C(v_i) \right\} \\ \geq \max \left[T_E(v_1) + \dots + T_E(m_i) + \dots + T_E(v_n), \max \{T_C(v_1), \dots, T_C(m_i), \dots, T_C(v_n)\} \right] \\ = T((v_1, \dots, m_i, \dots, v_n)) \quad (20)$$

仮定より, $T(v)$ は最小であるため, $T(v) = T((v_1, \dots, m_i, \dots, v_n))$ も最小となる. この定理によって, テスト集合 m_i より BIST を用いていないテスト集合はテスト時間最小化問題の解の候補から外して考えることができる.

□

定理 2 もし, テスト時間 $Time$ が達成できるのならば, $\exists v_i \in \{x \mid x \in V_i, T_C(x) \leq Time\}$ がテスト時間最小化問題の解を構成する.

証明 もし、テスト時間 $Time$ を達成できるのならば、 $\exists v_i \in \{x \mid T_C(x) > Time\}$ がテスト時間最小化問題の解を構成するとしてよう。

$$T = \max \left\{ \sum_{i=1}^n T_E(v_i), \max_{i=1}^n T_C(v_i) \right\}$$

$$\geq \max_{i=1}^n T_C(v_i)$$

$$> Time$$

これは矛盾である。

□

定理 3 もし、あるコア i において与えられたテスト集合のなかで最も BIST を用いるテスト集合 v のテスト時間が $\max_{i=1}^n T_C(m_i)$ 以下ならば、そのテスト集合はテスト時間最小化問題の解を構成する。

証明 最適解によるテスト時間を $Time$ とするとき、仮定と補題 1 より、コア i の任意のテスト集合 v_i において以下の式が成り立つ。

$$T_C(v_i) \leq \max_{i=1}^n T_C(m_i) \leq Time \quad (21)$$

この式より、コア i の任意のテスト集合 v_i が $T_{MAX}(v)$ を決定づけることはない。また、 $T_{SUM}(v)$ を最小にするコア i のテスト集合は四番目の仮定より、最も BIST を用いるテスト集合である。

□

5.4 アルゴリズム

本節では 5.2 節の仮定の下で効率の良いテスト時間最小化アルゴリズムを示す。定理 1 より、コア i において探索の候補となるテスト集合は m_i が m_i より BIST を用いるテスト集合である。初期解はこの条件を満たすテスト集合ベクトル v となる。アルゴリズムの制御構造は非常の単純なもので、 $\sum_{i=1}^n T_E(v_i)$ と $\max_{i=1}^n T_C(v_i)$ の大小関係によって決定される。 $\sum_{i=1}^n T_E(v_i)$ が $\max_{i=1}^n T_C(v_i)$ より大きいとき、 $\sum_{i=1}^n T_E(v_i)$ を削減するために、より BIST を用いているテスト集合が探索される。 $\sum_{i=1}^n T_E(v_i)$ が $\max_{i=1}^n T_C(v_i)$ より小さいとき、 $\max_{i=1}^n T_C(v_i)$ を削減するために、より外部テストを用いているテスト集合が探索される。 $\sum_{i=1}^n T_E(v_i)$ と $\max_{i=1}^n T_C(v_i)$ が等しいとき、以上の二つが行われる。

6. 実験結果

本節で、最小テスト時間と計算時間を示す。テスト時間最小化アルゴリズムを動作させる上で、Intel Pentium Pro 200MHz のプロセッサを搭載した、主記憶 64MB の PC を用いた。アルゴリズムは C 言語によって実装した。

Test Time Minimization Algorithm

Procedure Minimize (v)

Input : $v = (v_1, v_2, \dots, v_n)$

Output : The optimal test sets

begin

Remove unnecessary test sets
by Theorem 1 and 3.

repeat

$T := T(v)$;

Remove unnecessary test sets by Theorem 2.

if $\sum_{i=1}^n T_E(v_i) > \max_{i=1}^n T_C(v_i)$ then

Search test sets to reduce time of
external test part.

else if $\sum_{i=1}^n T_E(v_i) < \max_{i=1}^n T_C(v_i)$ then

Replace the test sets whose test time is
equal to T with less BISTed ones.

else /* $\sum_{i=1}^n T_E(v_i) = \max_{i=1}^n T_C(v_i)$ */

Replace the test sets whose test time is
equal to T with less BISTed ones.

And replace the other test sets with the
ones whose test time is the maximum
among the ones whose test time is less
than T .

endif

if test time is reduced then

Set new vector of test sets to v .

else

return v ;

endif

until the optimal test sets are found

end

図 8 テスト時間最小化アルゴリズム

表 3 ISCAS'85 ベンチマーク回路の諸元

回路名	セル数	入力数	出力数	サイクル数		故障数
				制御	観測	
C432	157	36	7	2	1	513
C499	202	41	32	2	1	750
C880	383	60	26	2	1	942
C1355	546	41	32	2	1	1566
C1908	878	33	25	2	1	1862
C2670	961	157	64	5	2	1990
C3540	1620	50	22	2	1	3126
C5315	2298	178	123	6	4	5252
C6288	2399	32	32	1	1	7638
C7552	3397	206	107	7	4	7041

表 4 ISCAS'85 ベンチマーク回路におけるテスト集合

回路名	C432	C499	C880	C1355	C1908	C2670	C3540	C5315	C6288	C7552
テスト集合数 (非理想)	30	46	33	83	96	27	95	63	18	66
テスト集合数 (理想)	29	41	23	68	53	19	60	47	14	61
1 テストパタン当りのサイクル数	外部	2	2	2	2	5	2	6	1	7
	BIST	1	1	1	1	1	1	1	1	1

10 個の ISCAS'85 ベンチマーク回路を用いた．回路の諸元を表 3 に示す．これらの回路を用いて，コア数が 10 から 20 の仮想的なコア・ベース LSI を仮定し，約三千万種類のコア・ベース LSI のテスト時間の最小化を行った．各コアのテスト集合数と外部テストの 1 テストパタンを適用するのに要するクロックサイクル数を表 4 に示す．非理想のテスト集合数とは 5.2 節の仮定を満たさないテスト集合の数である．理想のテスト集合数とは非理想のテスト集合を 5.2 節の仮定を満たすようにしたものである．本実験では BIST の周波数と外部テストの周波数 F_E をそれぞれ 32.0MHz と 6.4MHz とした．

アルゴリズムの CPU 時間と外部テストに対する CBET テスト手法のテスト時間の削減率を表 5 に示す．CPU 時間には非理想のテスト集合を 5.2 節の仮定を満たすようにする理想化の計算時間も含まれる．実験全体の計算量が非常に多いために，算出された CPU 時間は精度が悪いものとなっている．最悪の場合，CPU 時間は 0.41 秒である．20 個のコアによって構成されるシステムのテスト時間の最小化に要する CPU 時間の平均は 4.21 ミリ秒である．

本手法は 37.1% から 97.4% のテスト時間の削減を達成した．より多くのコアが並列にテストされれば，より多くのテスト時間が達成される．テスト時間の削減が小さい場合は，C2670 の回路がシステムに含まれており，最小化されるテスト時間は 600 マイクロ秒である．C2670 に与えられたテスト集合によるテスト時間

は 600 マイクロ秒あたりで解像度が悪く，それゆえに最小化がうまく行われなかったと推測できる．

そのような場合において，本アルゴリズムによる解は他の場合と較べてテスト時間の削減効果が低いものになることが理解される．本アルゴリズムは近似アルゴリズムであるが，本アルゴリズムによって得られた解と 5.3 節の定理を考慮した全探索を行えば，容易に最適解を得ることができる．

7. おわりに

本稿では，システム LSI 時代の要素技術となる新たなテスト手法である CBET テスト手法について議論した．テスト時間について，外部テスト，BIST，および CBET テスト手法のそれぞれを理論的，かつ実験的に解析した．本解析により，CBET テスト手法は他の手法に較べてテスト時間を小さくすることができることが明らかになった．さらに，CBET テスト手法をコア・ベース設計に拡張することによって，さらなるテスト時間削減が可能となった．CBET テスト手法は最高 97% 以上のテスト時間を削減し，LSI テスタ上のピンメモリサイズも削減した．テスト時間最小化を行う上で非常に効率の良いアルゴリズムについて議論した．

本研究の今後の課題として，CBET テストのためのテスト容易性尺度に関する研究，CBET テストにおけるテスト生成に関する研究，テスト時間とハードウェアのトレードオフを考慮したテストアーキテクチャに関する研究，および，コア・ベース LSI における最適なテスト集合を生成する研究が挙げられる．CBET テスト手法に関する研究は始まったばかりであり，今後の課題は多岐にわたる．しかしながら，課題を解決することによって得られるメリットは大きいものであり，CBET テスト手法は来世紀の半導体産業の主要技術の一つとなると筆者らは考える．

過去に日本はメモリの出荷において，徹底的な検査をよって拒否率を下げるといった方法で成功を収めた．台頭する諸外国の製品との差別化が困難な現在，企業が成功するためには何らかの差別化を製品に図らなければならない．高速な LSI テスタは非常に高価である

表 5 CPU 時間と削減率

コア数	CPU 時間 [ms]		削減率 [%]		
	最大	平均	最小	最大	平均
10	10.0	1.72	37.1	96.0	76.3
11	20.0	1.93	37.1	96.3	76.8
12	20.0	2.13	37.1	96.4	77.3
13	20.0	2.34	37.1	96.5	77.7
14	20.0	2.59	37.1	96.7	78.0
15	20.0	2.82	37.1	96.9	78.3
16	30.0	3.07	37.1	97.1	78.5
17	30.0	3.31	37.4	97.3	78.7
18	40.0	3.58	37.7	97.3	78.9
19	40.0	3.85	38.0	97.4	79.0
20	410.0	4.21	38.3	97.4	79.2

が、安価で低速な LSI テスタによってテストを行っても問題が生じない CBET テスト手法は非常に有効な商品の差別化の一つであると言える。

参 考 文 献

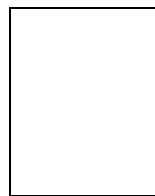
- 1) Bhatia, S., Gheewala, T. and Varma, P.: A Unifying Methodology for Intellectual Property and Custom Logic Testing, *Proc. Int. Test Conf.*, pp.639-648 (1996).
- 2) Touba, N. A. and Pouya, B.: Using Partial Isolation Rings to Test Core-Based Designs, *Using Partial Isolation Rings to Test Core-Based Designs*, pp.52-59 (1997).
- 3) Pouya, B. and Touba, N. A.: Modifying User-Defined Logic For Test Access to Embedded Cores, *International Test Conference*, pp.60-68 (1997)
- 4) Touba, N. A. and Pouya, B.: Testing Embedded Cores Using Partial Isolation Rings, *Proc. VLSI Test Symposium*, pp.10-16 (1996)
- 5) Kajihara, S, Pomeranz, I., Kinoshita, K. and Reddy, M.: Cost Effective Generation of Minimal Test Sets for Stuck at Faults in Combinational Logic Circuits, *Proc. the Design Automation Conf.* (1998).
- 6) Zorian, Y., Marinissen, E. J. and Dey, S.: Testing Embedded-Core Based System Chips, *Proc. International Test Conference*, pp.130-143 (1998).
- 7) 杉原 真: コアによって構成されるシステム LSI のテスト手法, 修士論文, 九州大学大学院システム情報科学研究科 (1998).
- 8) Sugihara, M., Date, H. and Yasuura, H.: Analysis and Minimization of Test Time in a Combined BIST and External Test Approach, *Proc. of Design, Automation and Test in Europe*, pp. 134-140 (2000).
- 9) Sugihara, M., Date, H. and Yasuura, H.: A Novel Test Methodology for Core-Based System LSIs and a Testing Time Minimization Problem, *Proc. of International Test Conference*, pp. 465-472 (1998).
- 10) Sugihara, M., Date, H. and Yasuura, H.: A Test Methodology for Core-Based System LSIs, *IEICE Trans. Fundamentals*, pp. 2640-2645 (1998).
- 11) Chang, J. S. and Lin, C. S.: Test Set Compaction for Combinational Circuits, *IEEE Trans. on Computer-Aided Design* pp. 1370-1378 (1995).

- 12) Schnurmann, H. D., Lindbloom, E. and Carpenter, R. F.: The Weighted Random Test-Pattern Generator, *IEEE Trans. on Computers*, C-24, No. 7, pp. 695-700 (1975).
- 13) Chakrabarty, K.: Test Scheduling for Core-Based Systems, *Proc. of International Conference on Computer Aided Design*, pp. 391-394 (1999).
- 14) Aerts, J. and Marinissen, E. J.: Scan Chain Design for Test Time Reduction in Core-Based ICs, *Proc. of International Test Conference*, pp. 448-457 (1998).
- 15) Hamzaoglu, I. and Patel, J. H.: Test Set Compaction Algorithms for Combinational Circuits, *Proc. of Int. Conf. on Computer Aided Design*, pp. 283-289 (1998).

(平成 ? 年 ? 月 ? 日受付)

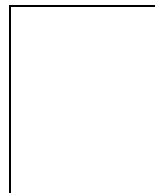
(平成 ? 年 ? 月 ? 日採録)

杉原 真



昭和 49 年生。平成 8 年九州大学工学部情報工学科卒業。平成 10 年九州大学システム情報科学研究科情報工学専攻修士課程修了。同年、富士通研究所(株)入社。平成 11 年より、九州大学システム情報科学研究科情報工学専攻博士後期課程在学。VLSI システムの設計手法と CAD の研究に従事。IEEE 学生会員。

安浦 寛人(正会員)



昭和 28 年生。昭和 53 年京都大学工学研究科修士課程(情報工学専攻)修了。京都大学工学部電子工学科助教授を経て、平成 3 年より九州大学大学院総合理工学研究科情報システム学専攻教授。現在、九州大学大学院システム情報科学研究院情報工学部門教授。九州システム情報技術研究所非常勤研究室長を兼務。VLSI システムの設計手法と CAD の研究に従事。平成 4 年情報処理学会論文賞、平成 5 年情報処理学会坂井記念特別賞および Best Author 賞をそれぞれ受賞。情報処理学会理事(基幹論文誌編集委員長)。ASP-DAC 2001 プログラム委員長。