

DCSTP 適用回路に対する最適電力テスト手法

A Suitable Power Test Method for DCSTP Circuit

小河 亮[†] 岩田 大志[‡] 山口 賢一[‡]
 Ryo Ogawa Hiroshi Iwata Ken'ichi Yamaguchi

1 はじめに

VLSI の出荷テストを行う際に、熱による破壊や過剰テストを防ぐためには通常動作時と同程度の電力消費とする必要がある。テスト時の消費電力が通常動作時よりも過大であると、熱による VLSI の破壊や、IR ドロップによって動作遅延が発生し、過剰テストとなる。一方で、テスト時の消費電力が通常動作時よりも過小であると、出荷後の品質を保証できない。本稿では DCSTP (Deterministic Circular Self-Test Path) [1] を適用した回路に対して、適切な電力でテストできる手法を提案する。

DCSTP はクロック毎にテストパターンを印加できるテスト手法であり、遅延故障のためのテストにも利用できる。また、スキャン動作を伴うテスト手法と比べてテスト実行時間が短いなどの利点がある。スキャン設計に対するシフト電力やキャプチャ電力などのテスト時電力の削減手法は多数提案されている [2, 3]。しかし、DCSTP を利用した消費電力削減手法はなく、類似手法を適用した回路に対する低消費電力テスト手法 [4] が提案されているのみである。その手法も大規模回路に対しては実用的な時間で適用できず、100% の故障検出効率を保証できないという問題がある。そこで本稿では DCSTP によって故障検出効率が 100% になることを保証しつつ、適用した回路に対して適切な消費電力でのテストを可能とする手法を提案する。

2 DCSTP

DCSTP は FF と組合せ回路部からなる順序回路を対象としている手法であり、出力応答を圧縮しながら組合せ回路部に適切なテストパターンを印加する手法である。

具体的な手法は、まず、順序回路に含まれる FF を図 1 のような DCSTP セルに置換し、着目する DCSTP セルの *pre_CELL* を前段の DCSTP セルの *suc_CELL* と接続する。さらに、着目する DCSTP セルの *suc_CELL* を後段の DCSTP セルの *pre_CELL* と接続することで

環状の FF チェーンを構築する。この FF チェーンを DCSTP チェーンという。ここで、*FI* は機能入力であり、*FO* は機能出力である。DCSTP チェーンは *B0* と *B1* で通常モードとリセットモード、テストモード、シフトモードを切り替える。リセットモードはテスト開始時に FF の状態を 0 にリセットするために用いられる。

テストモード時は、DCSTP チェーンは一つの長い MISR (Multiple Input Signature Register) となる。DCSTP における MISR は応答を圧縮するだけでなく、クロックごとにテスト応答を用いてテストパターンを生成し、組合せ回路部にテストパターンを印加する役割も担う。また、シフトモード時は DCSTP チェーンはシフトレジスタとなり、MISR で圧縮されたシグネチャを外部まで伝搬し、期待値との比較を行う。

図 2 にテスト時のチェーンの動作例を示す。PI と PO はそれぞれ外部入出力である。図 2 の Sel_i が 0 のとき *B* の XOR は *A* の XOR の出力をそのまま出力し、 Sel_i を 1 にすると、*B* の XOR は入力を反転させ、反転した値が FF に取り込まれ FO_i の出力となる。そのため、適切なセルの Sel_i を 1 にすると FO_i の値を反転させて印加できるため、組合せ回路部に適切なテストパターン印加できる。また、PI と PO はバウンダリスキャンを併用することで、制御・観測が可能となり、組合せ回路部に対して故障検出効率 100% を保証できる。 Sel_i

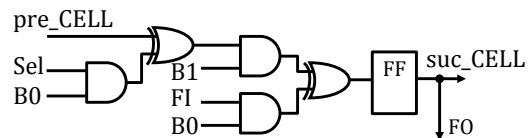


図 1: DCSTP セル

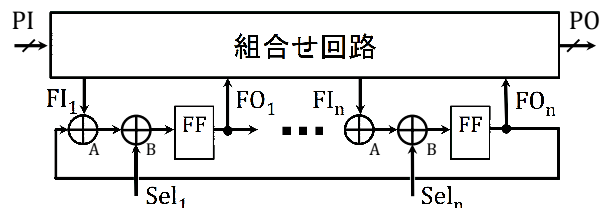


図 2: DCSTP チェーンの動作例

[†]奈良工業高等専門学校 電子情報工学専攻

[‡]奈良工業高等専門学校 情報工学科

はカウンタ回路とデコーダ回路によって制御する。カウンタは印加されたクロック数を記憶し、デコーダはカウンタの値をもとに特定の Sel_i の値をアクティブにする。

3 提案手法

本節では DCSTP に対して適切な消費電力でテストする手法を提案する。提案手法では回路全体の消費電力は FF の遷移回数に比例すると仮定し、テスト時に印加するテストパターン間の遷移するビットの数、つまり FF の遷移回数を制御することで、テスト時の消費電力を制御することを基本方針とする。また、遷移回数を通常動作時程度に制御することで、テスト時の消費電力も通常動作時程度にすることができる。

3.1 テストパターン順序付け問題

提案手法ではテストパターン集合 T と許容する遷移ビット数であるしきい値 th を入力し、 T と th から th に基づいた消費電力でテストするための順序付きテストパターン集合 T_p を求める。 T と th をもとに T_p を求める問題を **テストパターン順序付け問題** と定義する。また、その際テストパターンの追加操作を許すものとする。

定義 1 (テストパターン順序付け問題)

- 入力** ● T : テストパターン集合
 ● th : しきい値
- 出力** ● T_p : しきい値 th で定めた程度の消費電力でテストするための順序付きテストパターン集合
- 制約** ● テストパターンの追加操作を許す

本稿では問題を解くためにテストパターン集合 $T = \{t_1, t_2, \dots, t_n\}$ をグラフ $G = (V, E)$ を用いて、テストパターン間の関係を表現する。頂点集合 $V = \{v_1, v_2, \dots, v_n\}$ は t_i をラベルとして持つラベル付き頂点である。辺 $e = (v_i, v_j) \in E$ は v_i, v_j の持つラベル t_i, t_j 間のハミング距離がしきい値以下である場合に存在する。テストパターンのグラフ化の例を図 3 に示す。図 3 の例では 5 つのテストパターンがあり、しきい値は 2 となっている。

3.2 順序付けアルゴリズム

本稿で提案するテストパターン順序付け問題を解くアルゴリズムでは、グラフ G においてすべての頂点をたどるような経路 R を貪欲法により解を求める。提案アルゴリズムのフローチャートを図 5 に示す。アルゴリズムへの入力、 T としきい値 th であり、出力はしきい値 th で定めた程度の消費電力でテストするための順序付きテストパターン T_p である。

提案手法ではまず、入力 $T = \{t_1, t_2, \dots, t_n\}$ をラベルとして持つ頂点集合 $V = \{v_1, v_2, \dots, v_n\}$ を作成し、

$v_i, v_j \in V (i \neq j)$ 間のハミング距離がしきい値 th 以下の頂点間に辺を張ることでグラフ $G = (V, E)$ を構成する。

次に経路 R を探索する際の始点として、 $ALL - 0$ をラベルに持つ頂点を選択する。これは、DCSTP を適用した場合のテストはリセット状態、つまりすべての FF の値を 0 にしてから開始するためである。もし、 $ALL - 0$ をラベルに持つ頂点がなければ追加する。

以降は、すべての頂点を辿り終えるまで、経路 R の探索をする。次に辿る頂点は、現在着目している頂点に接続されている頂点の中から辞書順で上にくるものから選択する。頂点集合の中に未到達頂点が存在するにもかかわらず、次の未到達頂点への経路が存在しなければ、現在着目している頂点と最もハミング距離が近い未到達頂点を探索して、その頂点との間にしきい値を満たすように頂点を追加し、辺を張る。その後、形成された経路を辿り、未到達頂点の探索を続ける。このとき、しきい値によっては複数の頂点を追加する必要がある。

すべての頂点を辿り終えたあとで、経路 R を構成する頂点 v の順番と対応するようにテストパターンを順序付け、それをテストパターン T_p とする。最後に、テストパターン T_p を出力する。このテストパターン T_p を利用することで、通常動作時程度の電力でテストすることができる。

図 3 に対するテストパターンの順序付けの例を図 4 に示す。図 4 では頂点 000000 を始点とし、 $101000 \rightarrow 001000 \rightarrow 101010$ と順序付けていることを示してい

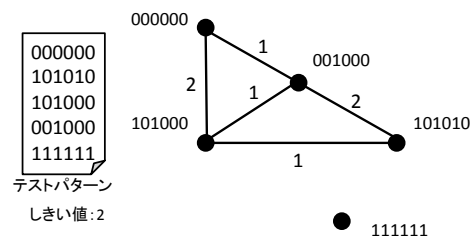


図 3: テストパターンのグラフ化

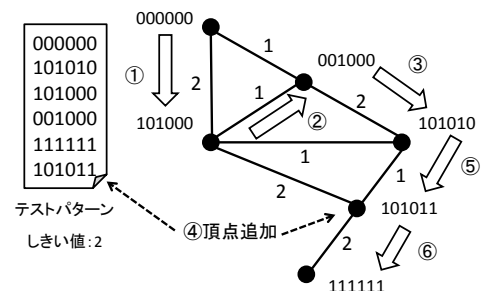


図 4: テストパターンの順序付け

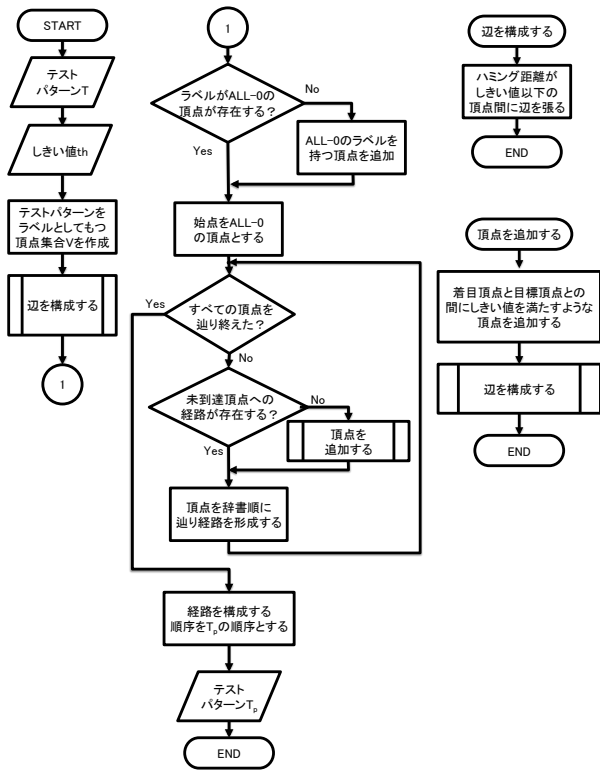


図 5: 提案手法のフローチャート

る。しかし、101010 と 111111 をつなぐ辺が存在しない。そのため、新たにテストパターン 101011 をラベルに持つ頂点を追加し、101010 から 111111 へ到達する経路を作成する。最終的にテストパターンは 000000 → 1010000 → 001000 → 101010 → 101011 → 111111 と順序付けられる。

3.3 提案手法を用いたテストフロー

提案手法を用いたテストフローは図6のようになり、3つの手順からなる。

1. 対象回路の組合せ回路部に対してテストパターンを生成する
2. 提案手法を用いてテスト時電力を考慮したテストパターンを順序付ける
3. 順序付けたテストパターンを印加できるようにDCSTPを適用する

手順1.では故障検出効率100%のテストパターン集合を生成する。手順2.で生成されたテストパターンを3.2節で述べたアルゴリズムを用いて適切な消費電力となるように順序付け、手順3.ではDCSTPを適用し順序付けたテストパターンを印加可能にする。以上の3つの手順で適切な消費電力が発生するようなテストができる。

4 実験方法

本稿では提案手法を用いてテストを行う場合と提案手法を用いずにテストを行う場合の消費電力を比較し、

提案手法の有用性を示す。

本稿で利用する評価指標は最大トグル率と平均トグル率、テスト実行時間である。トグル率は式(1)で表される。

$$\text{トグル率} = \frac{\text{toggle}(t_{i-1}, t_i)}{\#bit} \quad (1)$$

ただし、 $\text{toggle}(t_{i-1}, t_i)$ は $i-1$ 番目のテストパターン t_{i-1} から i 番目のテストパターン t_i に遷移したときに反転したビットの数を表し、 $\#bit$ はテストパターンのビット数を表す。最大トグル率は連続するテストパターン間のトグル率の最大値であり、大きいほど瞬間消費電力が大きいことを表す。平均トグル率は連続するテストパターン間のトグル率の平均値であり、平均的な消費電力を表す。テスト実行時間は、テストパターンの数である。DCSTPではクロックごとにテストパターンを印加するため、テストパターンの数がテスト実行時間と等しい。

本稿では、通常動作時程度のトグル率がテストパターンのビット数の20%が適切であると仮定し、提案手法に入力するしきい値をテストパターンのビット数の20%とした。本手法の有用性を評価するために、ISCAS'89のベンチマーク回路を用いた実験結果を表1に示す。表1の $\#bit$ はテストパターンのビット数を表している。実験環境は、プロセッサ: Intel Core i5 1.7 GHz, メモリ: 4GBである。

5 実験結果

表1より、提案手法を適用した場合は平均トグル率が通常動作時程度のトグル率であると仮定した20%に近い16.7%に抑えることができた。一方で、提案手法を適用しなかった場合は45.4%であった。また、最大トグル率の平均については、提案手法を適用した場合は、通常動作時程度の20.8%に抑えることができた。

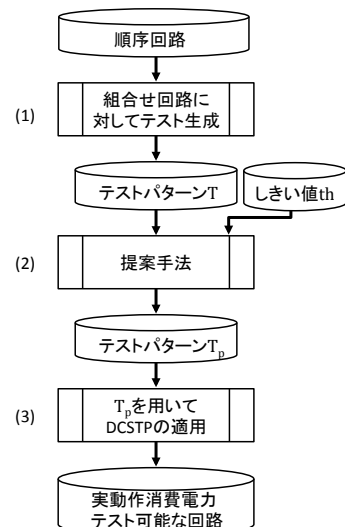


図 6: 適切な消費電力でテストするためのフロー

表1: トグル率とテスト実行時間の比較

適用回路	#bit	しきい値	提案手法なし			提案手法を使用した場合		
			最大トグル率	平均トグル率	テスト実行時間 (clock)	最大トグル率	平均トグル率	テスト実行時間 (clock)
s27	4	1	75.0%	40.0%	5	25.0%	21.4%	7
s298	16	4	81.3%	50.0%	7	25.0%	18.8%	13
s344	15	3	66.7%	43.7%	9	20.0%	15.9%	18
s382	23	5	69.6%	47.2%	7	21.7%	15.9%	15
s420	21	5	61.9%	34.5%	34	23.8%	18.6%	46
s510	16	4	68.8%	43.8%	37	25.0%	19.9%	55
s526	59	12	62.7%	37.9%	9	25.4%	16.9%	18
s641	49	10	63.3%	45.4%	9	20.4%	15.1%	23
s1196	19	4	57.9%	42.7%	18	21.1%	16.7%	33
s1423	88	18	68.2%	47.9%	11	20.5%	16.9%	25
s5378	208	42	53.4%	46.6%	29	20.2%	14.6%	84
s9234	235	47	62.6%	46.8%	20	19.6%	15.1%	57
s13207	691	139	64.7%	48.3%	30	17.4%	15.5%	88
s15850	600	120	74.5%	49.6%	20	17.3%	15.4%	58
s35932	1763	353	88.9%	48.8%	10	17.1%	15.6%	28
s38417	1664	333	70.3%	49.1%	31	17.1%	15.8%	91
s38584	1464	293	92.5%	49.2%	34	16.5%	15.8%	100
平均			69.5%	45.4%	18.8	20.8%	16.7%	44.6

しかし、提案手法を適用しなかった場合には通常動作時を大きく上回る 69.5% であった。

テスト実行時間は最小では s27 に適用したときの 2 パターンの増加で、最大では s35854 に適用したときの 66 パターンの増加であり、全体の平均では 25.8 パターン増加した。

また、提案手法を各ベンチマーク回路のテストパターンに対して適用して並び替えたときの処理時間はすべての場合において 1 秒未満であった。

6 まとめ

本稿では VLSI の出荷テスト時の熱の問題や IR ドロップによる動作遅延の問題を解決するために、適切な消費電力でテストできる手法を提案した。提案手法では連続するテストパターンをしきい値によって順序付け、順序付けたテストパターンを DCSTP を用いて印加することで、しきい値で定めた電力負荷で VLSI をテストすることを可能とした。実験結果から、提案手法を用いない場合では平均トグル率、最大トグル率ともにしきい値として設定した通常動作時の遷移を大きく上回っていたが、通常動作時程度のトグル率をしきい値として与え、提案手法を用いることで通常動作時程度に抑えられることを示した。また、その際のテスト実行時間は s38584 に適用したときが最大で 66 パターン増加し、s27 に適用した時が最小で 2 パターン増加した。

今後の課題は、提案手法の経路探索における辺を選択するとき、辞書順で選択するのではなく辺の先に

接続されている頂点の次数を用いた評価関数による優先順位にもとづいて選択し、テストパターンの増加を抑えることである。次数が少なければ到達できる可能性が少ないとして、より優先順位を高くすることが考えられる。また、テストパターン T_p のトグル率が通常動作時のトグル率の上限を下回らないように、つまり、テスト不足とならないように、下限を考慮した順序付けを行うことが考えられる。

参考文献

- [1] Ke WEN, Yu Hu, and Xiaowei LI, "Deterministic Circular Self Test Path," *Tsinghua Science and Technology Issue:S1*, Vol.12, pp.20–25, 2007.
- [2] Kohei Miyase, Kenji Noda, Hideaki Ito, Kazumi Hatayama, Takashi Aikyo, Yuta Yamato, Hiroshi Furukawa, Xiaoqing Wen, and Seiji Kajihara, "Effective IR-drop Reduction in At-Speed Scan Testing Using Distribution-Controlling X-Identification," *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*, pp.52–58, 2008.
- [3] Xiaoqing Wen, Kohei Miyase, Tatsuya Suzuki, Seiji Kajihara, Laung-Terng Wang, Kewal K. Saluja, and Kozo Kinoshita, "Low Capture Switching Activity Test Generation for Reducing IR-drop in At-Speed Scan Testing," *Journal of Electronic Testing: Theory and Applications*, Vol.24, No.4, pp.379–391, 2008.
- [4] F.Corno, M.Sonza Reorda, G.Squillero, and M.Violante, "CA-CSTP: A New BIST Architecture for Sequential Circuits," *Proceedings of the IEEE European Test Workshop*, pp.167–172, 2000.

本研究は東京大学大規模集積システム設計教育研究センターを通してシノプシス株式会社の協力で行われたものである。