

組合せ回路におけるテスト生成の効率化手法

AN EFFICIENT METHOD OF ATG FOR COMBINATIONAL CIRCUITS

宮沢 浩

Hiroshi MIYAZAWA

ソニー株式会社

SONY Corporation

Asahi-cho, Atsugi, 243 JAPAN

近年、ATGシステムの構成要素である検査入力生成、故障シミュレーションに対して数多くの試みがなされてきているが、ATGシステム全体の処理効率が考慮されることは少なかった。本論文では、検査入力生成と故障シミュレーションを統合した新しい検査系列生成手法を提案する。ここで述べる検査系列生成手法では、不要な処理を極力排斥することによって処理効率の向上がはかられ、さらに生成されるテストパターン数の最小化も考慮される。また、その有効性を評価するために行った実験の結果についても併せて報告する。

In recent years, many approaches have been made for test pattern generation and fault simulation which are the components of ATG system. However, the efficiency of ATG system, as a whole, have not been considered so much. In this paper, a new ATG method, in which test pattern generation and fault simulation are combined, is presented. With this method, unnecessary processes are rejected and minimization of test pattern is considered. The efficiency is also presented with experimental results.

1. まえがき

近年、LSIの高集積化に伴い、その検査系列の生成に要する時間、費用は増加の一途をたどっている。この問題を解決するために、順序回路のテスト生成問題を組合せ回路のそれに帰着させるスキャン・パス等のテスト容易化設計^{1,2}が広く用いられるようになり、組合せ回路の検査系列を効率よく生成するための試みが数多くなされている。しかし、それらの多くはATGシステムの構成要素である検査入力生成、故障シミュレーションに対するものであり、ATGシステム全体の処理効率が考慮されるこ

とはあまりなかった。

特に、検査入力生成アルゴリズムはその処理効率において飛躍的な発展を遂げながら、現在実用化されているATGシステムのほとんどが検査入力生成後に、ランダムパターンを付加して、故障シミュレーションを行うという古典的な手法を採用している。

そこで今回、検査入力生成と故障シミュレーションを統合した新しい検査系列生成手法を考案したので報告する。ここで述べる検査系列生成手法においては、ランダムパターンに代表される不確定要素及び不要な処理を極力排斥することによって処理効率の向上がはかられ、さら

に、生成されるテストパターン数の最小化も考慮される。

また、この新しい検査系列生成手法に基づいて開発された F D A S (Fault Detection And Simulation) システムによる実験結果を示しながら、その有効性について考察をする。

2. 不確定要素の排除

従来の一般的な A T G システムにおける検査系列生成の処理手順は、次の 2 つの Phase に大別することができる。

- 1) ランダムな検査入力生成と故障シミュレーション。
- 2) Algorithmic な検査入力生成と故障シミュレーション。

通常、2) の処理において検査入力生成の対象になる未検出故障は 1 つだけであり、Algorithmic に求められる入力信号値はわずかである。また、故障シミュレーション時には未設定の入力端子に対してランダムな信号値が割り当てられるため、残りの未検出故障が検出されるか否かはその多くが偶然性に委ねられてしまう。特に、入出力端子が多く論理段数の少ない回路においてはこの傾向が強い。

1) の処理においては、当然ではあるが、生成される検査入力には全て不確定要素のみで構成されるため、検出効率の低下を招き生成されるテストパターン数を増大させる要因になってしまう。

そこで、検出効率の高い検査入力を生成するためには、残りの未検出故障をできるだけ多く検出できるように経路を活性化することが望まれる。そのための方法として Dynamic Test Compaction (DTC) ³⁻⁴ が提案されている。

D T C では、「目標故障 F_i を検出するために求められた入力信号値を保持したまま、次の目標故障 F_{i+1} を検出する入力信号値を求める。」という処理が任意の回数繰り返されることによって検出効率の向上がはかられる。

また、故障シミュレーションにおいても不要

なイベントの発生を防ぐため、Algorithmic に活性化された経路上に仮定される未検出故障のみを扱えるようにすることが望ましい。

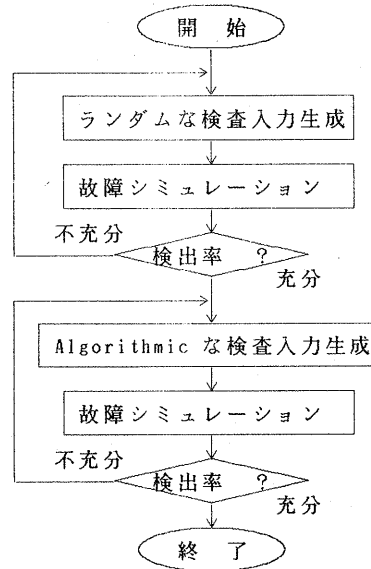


図 1. 一般的な A T G 処理フロー

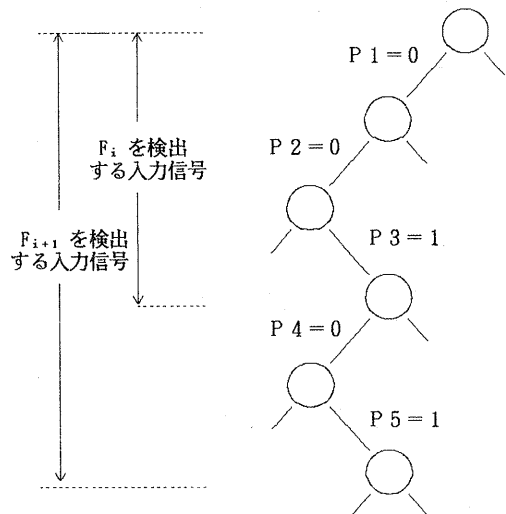


図 2. Dynamic Test Compaction

3. 検出対象故障の削減

検出対象故障を削減するのに有効なものとして、故障の等価性を用いて予め代表故障を定義しておくという方法が良く知られている。しかし、この方法では Concurrent,⁵ Deductive⁶ 等の手法による故障シミュレーションを行ったときに故障の支配関係が考慮されないため、削減方法として充分であるとはいえない。

例えば、図3に示される8個の故障は等価関係を用いることによって次の3つのグループに分けることができる。

- G1) A:SA0, B:SA0, C:SA1
- G2) D:SA1
- G3) E:SA0, F:SA1, G:SA1, H:SA0

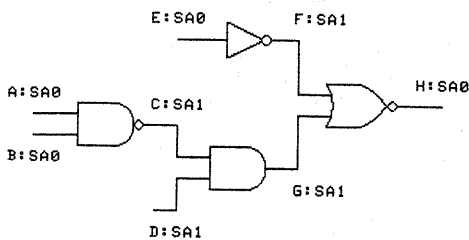


図3. 検出対象故障の削減

それぞれの等価故障グループに対して代表故障を定義したとしても最低3つの故障を検出対象としなければならない。しかし、仮に1つの代表故障のみを検出対象として故障シミュレーションを行うことができれば、グループ G1 のいずれかの故障が検出された時点で、C:SA1, G:SA1 の支配関係を用いることにより、グループ G3 の全ての故障も検出対象から除くことができる。

このように個々の故障に対して独立に故障シミュレーションを行う方法として Single Fault Propagation (SFP)⁷ がある。

SFPでは、「与えられた検査入力により内部状態が安定した回路内の任意の箇所」に故障を仮定し、前方へのドライブ操作を行って、その

影響が1次出力端子まで伝搬されるか否かを判定する。」という処理が仮定可能な全ての故障に対して行われる。

同様に、検査入力生成処理においても故障の等価性、支配性を用いることによって検出対象故障を効果的に削減することができる。

4. 効率的な検査系列生成

計算機上に構築されるATGシステムの処理効率を上げるためには、プログラム及び内部テーブルの構造を簡素化すると共にメモリ使用量を削減することも重要である。そこで、DTCとSFPを統合して検査入力生成と故障シミュレーションを同時に行う検査系列生成手法を考案した。以下にその処理手順を示す。

- S1) 全てのゲートの値を 'X' にする。
- S2) まだ検出を試みていない未検出故障 F1:SA_v を1次故障として選ぶ。そのような故障が無ければ終了。
- S3) 1次故障 F1:SA_v を検出する入力信号値を求める。検出することができなければ F1:SA_v と等価な全ての故障を 'UNDETECTABLE' として検出対象から除き S2 へ。
- S4) 等価性、支配性により検出が保証される全ての故障を 'DETECTED' として検出対象から削除。
- S5) 現在の入力信号値でまだ検出を試みていない未検出故障の中から、その信号線 F2 の現在の値が 'v' でない故障 F2:SA_v を2次故障として選ぶ。そのような故障が無ければ現在の入力信号値を検査入力として出力し S1 へ。
- S6) 現在の入力信号値を保持したまま2次故障 F2:SA_v を検出する入力信号値を求める。検出することができたら S4 へ。そうでなければ S5 へ。

F2:SAV が 2 次故障として検出不可能であると判定された場合、それと等価な全ての故障も 2 次故障として検出することが不可能である。しかし、1 次故障における 'UNDETECTABLE' とは本質的に異なるためこれらの故障を検出対象から除くことはできない。そこで、これらの故障に対しても 2 次故障としての検出を試みるという不要な処理が行われることを避けるため、S5 においては等価な故障集合の中から代表故障として定義されるただ 1 つの故障のみを 2 次故障として選ぶ。一方、1 次故障の検出においては、故障検出率の低下を防ぐため全ての故障を検出対象とし、バックトラック数、処理時間等の制限により F1:SAV の検出処理が打ち切られた場合には、F1:SAV のみを 'ABORTED' として検出対象から除くようにする。

S6 の処理において F2:SAV を検出するために新たな入力信号値が追加されれば DTC が行われたことになる。また、新たな入力信号値を追加することなく F2:SAV が検出されれば SFP が行われたことになる。このような処理は経路活性化法に基づく検査入力生成アルゴリズムを用いることにより、簡単なスタック操作で実現することができる。

本手法は以下のような特徴を持つ。

- 1) 検査入力生成と故障シミュレーションが同時に行われるため、従来の ATG システムで行われていた "ランダムな検査入力生成と故障シミュレーション" という処理が必要でなくなる。
- 2) 検査入力生成を効率良く行うことによって故障シミュレーションを実現するため、独立したモジュールとしての故障シミュレータは存在しない。従ってわずかなプログラムサイズで ATG システムを構築することができる。
- 3) 常に 1 つの故障のみを処理対象としているため伝搬故障リストのように大きなテーブルを必要とせず、実行時のメモリ使用量の予測が容易である。
- 4) 残りの未検出故障を 2 次故障として検出

する可能性が無くなった時点で経路活性化処理を打ち切るため、'X' (don't care) を含む形で検査系列が生成される。

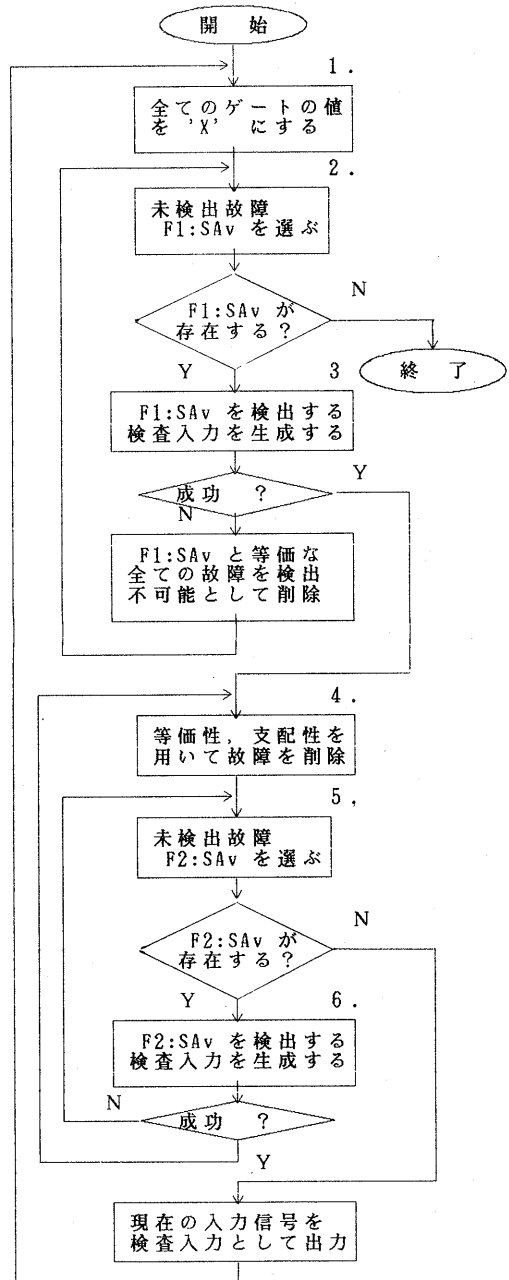


図 4. 内部処理フロー

5. 実行結果

前述の処理に基づいて開発された検査系列生成システムFDASによる実験を行い、その有効性を評価した。FDASでは検査入力生成部にPODEM⁸を採用し、可制御性、可検査性の尺度としてはそれぞれ入出力端子からのレベルのみを用いている。また、回路の状態を表す論理値には0, 1, X, D, \bar{D} にハイ・インバーダンス'Z'を加えた6値を使用している。

評価用データとして用いた回路⁹の特性を表1に示す。また、1次故障及び2次故障の検出におけるバックトラックの上限値を(10, 0), (1000, 0), (10, 5)とした場合の実行結果を表2~4に示す。BTK1, BTK2で示される値はそ

れぞれ1次故障及び2次故障の検出において発生したバックトラックの総数である。

なお、実験に使用した計算機はFACOM-VP50(15.5 MIPS)である。

表1. 回路特性

回路	ゲート数	入力/出力	故障総数
C432	160	36/7	864
C499	202	41/32	998
C880	383	60/26	1760
C1355	546	41/32	2710
C1908	880	33/25	3816
C2670	1193	233/140	5340
C3540	1669	50/22	7080
C5315	2307	178/123	10630
C6288	2416	32/32	12576
C7552	3512	207/108	15104

表2. 実行結果 (BACK TRACK LIMIT = 10, 0)

回路	検出故障数	削減数	打ち切り	BTK1	BTK2	検出率	検査数	CPU(秒)
C432	843	406	21	233	0	97.57	51	0.83
C499	990	251	8	87	0	99.20	102	1.28
C880	1760	992	0	2	0	100.00	38	0.69
C1355	2702	1474	8	154	0	99.70	131	2.94
C1908	3801	2187	9	106	0	99.61	144	3.43
C2670	5148	2895	155	1675	0	96.40	91	11.65
C3540	6792	4022	203	2386	0	95.93	148	34.03
C5315	10560	6028	26	515	0	99.34	132	44.54
C6288	12504	6614	8	223	0	99.43	34	17.27
C7552	14884	8662	144	1712	0	98.54	151	43.75

表3. 実行結果 (BACK TRACK LIMIT = 1000, 0)

回路	検出故障数	削減数	打ち切り	BTK1	BTK2	検出率	検査数	CPU(秒)
C432	853	408	11	12509	0	98.73	57	4.26
C499	990	251	8	8007	0	99.20	102	4.63
C880	1760	992	0	2	0	100.00	38	0.65
C1355	2702	1474	8	8074	0	99.70	131	8.46
C1908	3801	2187	9	9016	0	99.61	144	11.07
C2670	5148	2895	99	107586	0	96.40	91	65.75
C3540	6810	4026	131	139099	0	96.19	159	139.27
C5315	10568	6030	4	5298	0	99.42	132	50.37
C6288	12508	6616	4	6174	0	99.46	35	23.01
C7552	14885	8723	123	126961	0	98.55	152	149.58

表4. 実行結果 (BACK TRACK LIMIT = 10, 5)

回路	検出故障数	削減数	打ち切り	BTK1	BTK2	検出率	検査数	CPU(秒)
C432	843	406	21	232	5679	97.57	50	0.85
C499	990	251	8	87	4853	99.20	102	1.32
C880	1760	992	0	2	3492	100.00	38	0.80
C1355	2702	1474	8	154	12925	99.70	131	3.00
C1908	3801	2187	9	106	16300	99.61	144	3.86
C2670	5148	2890	155	1673	46299	96.40	89	25.49
C3540	6797	4026	198	2340	92063	96.00	143	58.38
C5315	10560	6031	26	510	106374	99.34	132	115.55
C6288	12507	6616	5	176	5540	99.45	31	17.81
C7552	14885	8718	143	1656	82775	98.55	143	58.68

C2670, C3540, C7552 等の回路に対しては1次故障の検出において多くのバックトラックが発生し、処理時間にも影響していることが明らかである。また、C5315 に対しては2次故障の検出において発生するバックトラックが多く、やはり処理時間に影響していることがわかる。これは、検査入力生成部に採用したPODEMではバックトラック数の削減が考慮されないためである。

本手法は検査入力生成アルゴリズムを限定するものではないので、例えば、仮定した故障の検出が不可能であることを早期に発見する工夫がなされたFAN¹⁰アルゴリズム等を用いることによって、さらに良い結果が得られるものと考えられる。逆に、バックトラックによるオーバーヘッドがほとんど無いC880, C6288等の回路に対する実行結果が本手法の持つポテンシャルの高さを示していると言えよう。

バックトラック数の削減と同様に、可制御性、可検査性の尺度や論理値等FDASには改善の余地が多く残されているにもかかわらず、何れの回路に対しても故障検出率及び検出効率の高い検査系列をわずかな処理時間で生成することができた。また、平均して約56%もの故障が等価性、支配性により検出対象から除かれており、本手法による検査系列生成処理が効率の良いものであることが確認された。

6. あとがき

本論文では検査入力生成と故障シミュレーションを統合し、生成されるテストパターンの最小化を考慮しながら処理効率の向上をはかる新しい検査系列生成手法を提案した。また、実験結果を用いてその有効性を示すと共に、実験に用いたFDASシステムにはさらに改良されるべき点があることを指摘した。

約2000ステップのFORTRAN 77で記述されたFDASは現在13M byteのテーブルで10万ゲート規模の回路に対応しており、パソコンやワークステーションといった小規模な計算機上での実行も充分可能であること等、実用上の様々な効果が期待される。

参考文献

1. E.B.Eichelberger and T.W.Williams, "A logic design structure for LSI testing", Proc. 14th Design Automation Conf., pp. 462-468, 1977
2. S.Funatsu, N.Wakatsuki, T.Arima, "Test generation system in Japan", Proc. 12th Design Automation Conf., pp.114-122, 1975
3. P.Goel, "Dynamic subsumption of test patterns for LSSD systems", IBM Tech. Discl. Bulletin, Vol. 21, No. 7, Dec. 1978, pp.2782-2784
4. P.goel, B.C.Rosales, "Dynamic Test Compaction with Fault Selection Using 'sensitizable' Path Tracing", IBM Tech. Discl. Bulletin, Vol.23, No. 5, Oct. 1980, pp.1954-1958
5. E.G.Ulrich, T.Baker, "The Concurrent Simulation of Nearly Identical Digital Networks", Proc. 10th Design Automation Workshop, pp.145-150, 1973
6. D.B.Armstrong, "A Deductive Method for Simulating Faults in Logic Circuits", IEEE Trans. Comput., Vol. C-21, No. 5, pp.464-471, 1972
7. F.Ozguner, W.E.Donath, C.W.Cha, "On Fault Simulation Techniques", Journal of Design Automation & Fault Tolerant Computing, Vol. 3, No. 2, Apr. 1979, pp.83-92
8. P.Goel, "An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits", IEEE Trans. Comput., Vol. c-30, pp.215-222
9. F.Brglez, H.Fujiwara, "A neutral Netlist of 10 combinational circuits", Special Session on ATPG and Fault Simulation, Proc. 1985 IEEE Int. Symp. Circuits and Systems, Kyoto, Japan, Jun. 1985
10. H.Fujiwara, "On the Acceleration of Test Generation Algorithms", IEEE, Trans. omut., Vol. c-32, pp1137-1144, Dec. 1983