

7V-4

論理シミュレーション専用マシンを用いた
テスト生成システムにおける高検出率化手法

高山 浩一郎 広瀬 文保
(株)富士通研究所

1.はじめに

我々は、論理シミュレーション専用マシンSP[1]を応用した、組合せ回路の単一縮退故障に対する高速な自動テスト生成システム[2]を研究中である。本システムでは、図1に示すように、被検査回路(組合せ回路)と仮定故障リストを入力して「テスト生成回路」をホスト上で自動合成し、SPを用いてテスト生成回路の動作を高速に模擬することによりテスト系列を自動生成する。

テスト生成回路のアルゴリズムには乱数を入力パターンとするモードがあり、これをランダムパターン(RP)型と呼ぶ。また、入力空間の縦型探索を行うモードがある。これは、現在検査対象となっている故障がある入力に対して検出不能であるとき、入力をバックトラックするか、入力をそのままにして故障を取り替えるかを選択でき、前者を故障先行(FF)型、後者を入力先行(IF)型と呼ぶ。

これまでに、RP型とIF型を実現して、その性能評価を行った。今回、さらに高い検出率を達成するためにFF型を実現し、アルゴリズムをRP型からFF型

に切り換えて実験を行い有効性を確認した。また、入力の探索順序を故障に応じて動的に変更することでさらに高い検出率が得られることがわかった。

2.故障先行 (FF) 型アルゴリズム

2.1.アルゴリズムの概要

FF型は一般に用いられているテスト生成手法である。まず、被検査回路中に故障を仮定し、それを検出する入力パターンを求める(テスト生成)。テストパターンが求めたならば、故障を順次取り替えて等価故障を検出する(故障シミュレーション)。この操作を、未検出故障が無くなるまで繰り返す。

2.2.ベンチマーク回路[3]による評価

実験では、最初にRP型を用いて故障を検出した後、未検出の故障に対してFF型を適用して性能を評価した。テスト生成回路は正常回路と2個の故障回路を同時にシミュレーションすることができる。故障は回路記述ファイルに出現する順に8個の集合に分割し、2個の集合を組にして故障挿入器を構成することで合成された4個のテスト生成回路を4台のプロセッサを用いて同時にシミュレーションした。

RP型では、ホスト上のシステム関数を用いて生成した乱数を用いた。連続する32個のパターンが故障を検出しなかったときRP型を終了した。FF型では、入力の探索中に、バックトラックの結果検出の可能性が不明となった回数を仮定故障ごとにカウントし、256回に達した故障を検査対象からアボートするようにした。結果を表1に示す。冗長数は、冗長性が証明された故障数を示す。この実験において、冗長性の証明された故障数が全体的に少なく、c2670やc7552のように依然として多くの故障がアボートする原因として、次の2つがあげられる。

(原因1)入力の探索順序がテスト生成回路合成時に固定されるため、特に入力数の大きい回路において、故障によっては無駄な空間を探索してしまう。

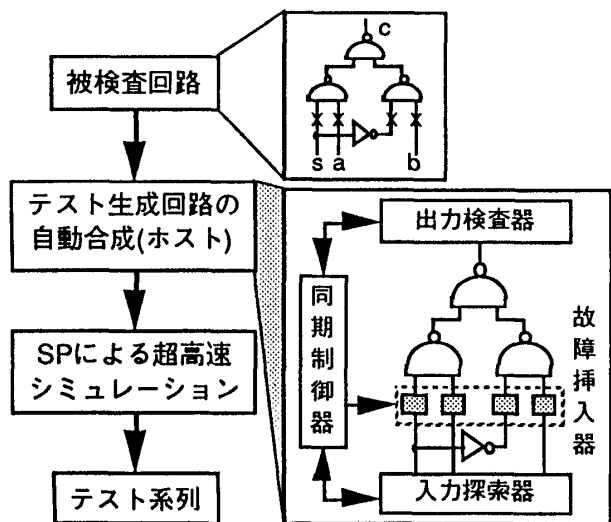


図1. テスト生成システムの概要

(原因2)テスト生成時に2個の故障を同時に挿入しているが、符号の解像度の限界により一方の故障のバックトラックを他方の故障もカウントしてしまう場合がある。

今回は、(原因1)に対処するために探索順序を故障に応じて動的に変更する方式を実現した。

3.高検出率化への一手法

3.1.探索順序の動的な変更

各故障に対して全入力を図2のように3つの集合に分割する。

- A.故障を仮定する信号線aへのバスが存在する入力。
- B. 信号線aから到達可能な出力へのバスが存在する入力。
- C. A,Bに含まれない入力。

この分割の結果を行列としてSP内部のメモリにロードし、入力探索器は探索時に仮定される故障に応じてメモリを参照して、A→B→Cの順に探索を行う。

3.2.ベンチマーク回路による評価

本方式の有効性を評価した。実験では、(原因2)の原因の影響を無くすために、RP型適用後の未検出故障をもとに、同時に高々1個の故障を挿入するテスト生成回路を再合成し、FF型を適用した。結果を表2に示す。時間比は、探索順を動的に変更した場合に要した時間を固定した場合の時間で割ったものである。本方式は、探索順序を固定した場合と比較すると、検出した故障数、冗長性を証明した故障数は

いずれも増加し、シミュレーション時間も短縮していることから、本手法の有効性がわかる。

4.おわりに

故障先行型による動作制御方式を実現してその有効性を確認した。また、故障に応じて入力探索順序を動的に変更することで、さらに検出率を向上できる可能性があることを確認した。現在、符号化法から生じる検出率の低下(原因2)を抑えるテスト生成モードを実現中である。

[参考文献]

- [1] F. Hirose, et al., "Simulation Processor "SP"", ICCAD 87, pp.484-487.
- [2] 広瀬, 高山, 川戸, "テスト生成回路を用いた高速テスト生成方式", 信学論(D1)採録済.
- [3] F. Brglez and H. Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortran", ISCAS 85.

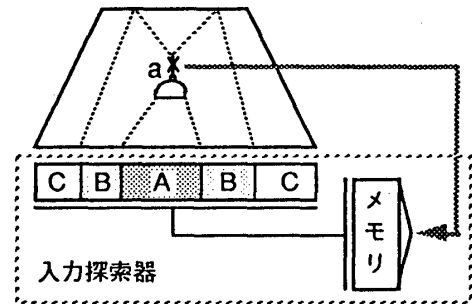


図2. 探索順序の動的な変更

表1. 故障先行型によるテスト生成実験結果

| 回路名 | 故障数 | RP型 | | FF型 | | | | 全体 時間(秒) |
|-------|------|------|--------|-----|-----|-------|--------|-------------|
| | | 検出数 | 時間(秒) | 検出数 | 冗長数 | アボート数 | 時間(秒) | |
| c2670 | 2747 | 2266 | 2.728 | 338 | 0 | 143 | 11.095 | 13.823 |
| c3540 | 3428 | 3225 | 5.511 | 61 | 50 | 92 | 5.102 | 10.613 |
| c5315 | 5350 | 5257 | 7.022 | 25 | 0 | 68 | 4.371 | 11.393 |
| c6288 | 7744 | 7710 | 9.974 | 0 | 0 | 34 | 6.113 | 16.087 |
| c7552 | 7550 | 6944 | 12.814 | 298 | 26 | 282 | 11.488 | 24.302 |

表2. 探索順の動的変更実験結果

| 回路名 | 対象 故障数 | 探索順固定 | | | 探索順動的変更 | | | 時間比 |
|-------|-----------|-------|-----|-------|---------|-----|-------|-------|
| | | 検出数 | 冗長数 | アボート数 | 検出数 | 冗長数 | アボート数 | |
| c2670 | 481 | 353 | 0 | 128 | 355 | 19 | 107 | 0.914 |
| c3540 | 203 | 62 | 50 | 91 | 63 | 81 | 59 | 0.720 |
| c5315 | 93 | 26 | 0 | 37 | 33 | 41 | 19 | 0.380 |
| c6288 | 34 | 0 | 0 | 34 | 0 | 32 | 2 | 0.063 |
| c7552 | 606 | 342 | 31 | 233 | 349 | 44 | 213 | 0.889 |