

乱数伝搬度に着目した BIST 評価手法

1 K-8

池永 剛† 高橋 淳一‡ 小倉 武†

NTT LSI 研究所† NTT ヒューマンインタフェース研究所‡

1 はじめに

高集積化が進む DSP-LSI に対するテスト容易化の一手法として、BIST (Built-In Self Test) があげられる^[1]。しかし、BIST は一般にテストパターン長が長くなるため、故障検出率算出等の評価は困難になりつつある。そこで、評価の効率化、低コスト化を可能とする BIST 評価手法を検討している。本稿では、DSP-LSI のコアである、演算器がネットワーク結合された DPU (Data Path Unit) に対する BIST 評価手法を提案し、実回路に適用した評価結果を示す。

2 従来の評価手法の問題点

BIST の評価手法(ここでいう評価とは、目標とする故障検出率を達成可能なテストパターン数を決めることと考える)として、従来から行なわれている故障シミュレータを用いて、故障検出率を算出する手法が考えられる。しかし、この手法を用いて、DPU に対する BIST の評価を行なう場合、次の問題点が考えられる。

- DPU 全体(十数 kG 規模)に対し、長大パターンの故障シミュレーションを行なう必要がある。このため、大規模回路、順序回路の扱える高価なハードウェアシミュレーションエンジンを長時間占有せねばならず、多大の計算機コストを必要とする。
- DPU 全体のネットリストを必要とするため、最終的な DPU のデータバスが決まるまでは BIST の評価が開始できない。また、データバスの変更、BIST 構成の変更が行なわれるたびに最初から評価をやり直す必要があり、非効率である。

3 BIST 評価手法

2 で述べた問題点を解決するため、DPU の構成要素である個々の演算器の評価と、演算器のネットワーク構成の評価をそれぞれ独立に行ない、それらの結果を照らし合わせて DPU 全体の BIST 評価を行なう手法を提案する。このうち、個々の演算器の評価は、故障シミュレータ、あるいは故障検出確率を与えてくれる STAFAN^[2]等を用いて、各演算器ごとに、乱数何パターンで十分な検出率が得られるかを調べる。また、ネットワーク構成の評価は、今回提案する、DPU の入力に与えた乱数が各演算器の入出力にどれくらい伝搬するかという点に着目したメジャを用いて評価を行なう。

An Evaluation Method for a Built-In Self Test using a Randomness Transmission Measure.

Takeshi IKENAGA†, Jun-ichi TAKAHASHI‡, Takeshi OGURA†
NTT LSI Laboratories†, NTT Human Interface Laboratories‡

この手法を用いた評価は、各演算器(数百～数千 KG 規模)の故障シミュレーション+ DPU の機能シミュレーションによって行なえ、計算機コスト削減が可能である。また、個々の演算器マクロが完成次第、順次評価が開始できるため、全体のデータバス設計と BIST 評価を並列に行なうことができ、効率的な評価が可能となる。

次に、ネットワーク構成の評価手法を述べる。

4 ネットワーク構成の評価

本稿では、評価対象を、LFSR(Linear Feedback Shift Register)を用いた BIST を前提として考える。この場合、テストパターンは、対象回路に無依存な乱数パターンとなるので、DPU のネットワーク構成の評価は、DPU 内におけるランダム性の伝搬度の評価に置き換えて考えることができる。このランダム性を表す指標としては、エントロピーを用いた例^[3]が報告されているが、全確率事象空間(全数パターン)におけるパターンの発生頻度を求める必要があり、多ビットデータバスにおけるエントロピーの算出は困難である。ここでは、DPU において容易に算出可能な指標として、パターン種別(パターン数)をランダム性の指標として取り上げる。

次に、ランダム性の伝搬度を示すメジャを考えるが、ここでは、Testability Measure 等の算出に用いられる、可制御、可観測の概念を用いる。つまり、DPU 内のある演算器に着目した場合、その演算器がテストされるためには、DPU の入力に与えた乱数パターンが演算器の入力に伝搬していること(可制御)と、演算器に生じた故障の影響が途中でマスクされずに DPU の出力に伝搬すること(可観測)が必要になるが、この考え方に従い、次の4パラメータからなるメジャを導入する。

$$RC(\text{乱数可制御性}) = \frac{\text{演算器の入力に現われるパターン数}}{\text{DPU の入力に与える乱数パターン数}}$$

$$RT(\text{乱数透過度}) = \frac{\text{演算器の出力に現われるパターン数}}{\text{演算器の入力に現われるパターン数}}$$

$$RO(\text{乱数可観測性}) = RT \text{ の積 (演算器出力} \rightarrow \text{DPU の出力)}$$

$$RM(\text{乱数伝搬メジャ}) = RC(\text{乱数可制御性}) \times RO(\text{乱数可観測性})$$

これらの式において、RC は、DPU の入力に与えたパターンが、異なるパターンとして各演算器に伝搬する確率、RO は、各演算器に生じた故障の影響が、途中でマスクされずに DPU の出力に伝搬する確率を示している。よって RM は、DPU 全体で BIST を構成した場合、DPU の入力に与えたパターンの何 % で、各演算器がテストされたかを示すメジャとみなせる。

これらの値を、予め、構成要素の評価として各演算器ごとに故障シミュレーション等で求めておいた、十分

な検出率を得るために必要なパターン数と照らし合わせ、DPU 全体では、何パターンでテストするかを決定する。

5 適用例

画像処理用 DSP^[4] の DPU 部に対して、上のメジャを用いた評価を行なう。図 1 に ALU に対するメジャ算出方法を示す。各メジャの値は、各演算器の入出力のパターン数の値で求められるので、DPU の機能記述を用いた機能シミュレーションによって算出できる。

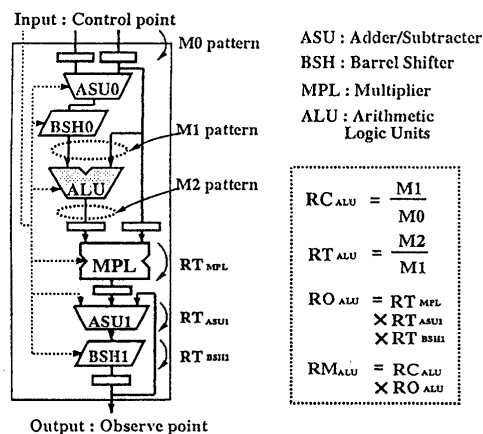


図 1: DPU 内の ALU に対する乱数メジャー

同様の方法により、全ての演算器に対して乱数伝搬メジャーを求めた (LFSR により乱数 2000 パターンを DPU の入力に与えて算出) 結果を表 1 に示す。

表 1. 乱数伝搬メジャー (2000 パターン)

演算器	RC	RT	RO	RM	有効パターン
ASU0	1.000	0.986	0.355	0.355	710
BSH0	0.986	0.806	0.441	0.435	870
ALU	1.000	0.901	0.489	0.489	978
MPL	0.994	0.962	0.509	0.506	1012
ASU1	1.000	0.986	0.516	0.516	1032
BSH1	0.986	0.516	1.000	0.986	1972

表 1 に示すように、RM は 0.3 ~ 0.9 を示している。この RM に 2000 をかけたパターン数が各演算器での有効パターン (単体では何パターンでテストしたことに相当するか) となる。これらの値は全般的に高くなっているが、DPU は単一フローのデータバス構成をとるため、順序回路の中でも、出力系列が入力にフィードバックする回路構成をとるものと比べて、ランダム性の伝搬度が高いためであると考えられる。

次に、この乱数伝搬メジャーの評価を行なう。上で述べたように、RM は、DPU 全体で BIST を構成した場合、DPU の入力に与えたパターンの何 % で、各演算器がテストされたかを示す。よって、RM が正確な指標を示しているとする、DPU 全体を 2000 パターンでテストした場合の各演算器における故障検出率と、演算器個々を $2000 \times RM$ パターンでテストした場合の故障検出率が一致するはずである。この DPU 全体と演算器個々の故障検出率の比較結果を表 2 に示す。

表 2. 乱数伝搬メジャーの評価

演算器	故障検出率 (%)	
	DPU (2000 パターン)	演算器 (2000 × RM パターン)
ASU0	98.83	98.72
BSH0	82.30	82.17
ALU	99.55	99.40
MPL	98.62	98.63
ASU1	96.14	95.99
BSH1	97.98	97.98

上の比較時において、故障検出率がまだ上昇しているもの、既に収束値に近付いているものなど、様々な過程のものが存在するが、故障検出率の差はいずれも $\pm 0.15\%$ 内に収まっており、故障シミュレーションの代替手段として十分な精度を持つことが確認できた。

評価の結果、現実的なパターン数 (テスト時間) では所望の故障検出率を得られない場合、BIST 構成に何らかの修正を加えていく必要がある。本手法では、個々の要素とそのネットワーク構成に分けて評価しているため、その修正は、比較的容易に行なうことができる。つまり、評価の結果、ネットワーク構成が乱数テストに向かないと判明した場合は、途中のパイプラインレジスタ等を BIST レジスタに置き換えて、可制御性、可観測性を向上させる。一方、個々の演算器が乱数テストに向かないと判明した場合は、観測点挿入等によって演算器の回路構成を乱数テスト向きに改善する。

この DPU の例では、RM の値は、最低でも 0.3 程度であり、ネットワーク構成は比較的 BIST に向いている。一方、個々の演算器において、BSH0 の故障検出率 (82%) が低いが、このような回路に対しては、観測点挿入等によって、より少ないパターン数で高い検出率が得られるような回路にする必要がある。

6 おわりに

本稿では、DPU に対する BIST 評価手法として、構成要素の評価とネットワーク構成の評価を分けた分割評価手法を提案した。また、ネットワーク構成の評価手法として、DPU の入力に与えた乱数が各演算器の入出力にどれくらい伝搬するかという点に着目したメジャーを提案し、実回路に適用した。この結果、本手法は、故障シミュレーターの代替手段として、評価の効率化、低コスト化が図れるという見通しを得た。今後は、多くの例題を用いた評価をおこない、さらに精度の高い評価手法への改良を行なっていきたい。

参考文献

- 池永、高橋：DSP-LSI に適した BIST 構成法、信学技報 FTS92-8、(1992)
- S.K.Jain, et al., "STAFAN: An Alternative to Fault Simulation," 21st Design Automat. Conf., pp.18-23 (1984)
- K.Thearling, et al., "An Easily Computed Functional Level Testability Measure," Int. Test Conf., pp.381-390 (1989)
- T.Minami, et al., "A 300-MOPS Video Signal Processor with Parallel Architecture," ISSCC, Digest of Technical Paper, Vol.34, pp.252-253 (1991).