

相互に接続されたPLAに対する 並列故障シミュレーション

3H-8

庄司隆夫[†] 森美紀[†] 松崎英樹[‡] 井口幸洋[†] 山田輝彦[†]
[†]明治大学理工学部情報科学科 [‡]明治大学情報科学センター

1. はじめに

PLA(Programmable Logic Array)は規則的な構造ゆえ論理回路の設計の自動化が容易なことから広く用いられている。単一のPLAについては入力パターンのビット表現とPLA構造の規則性を活用した並列ビット演算を基に、各テストパターンで検出できるクロスポイント故障を演繹して高速化をはかったシミュレーション手法が提案されている[1]。しかし、この手法を相互に接続されたPLAにそのまま適用することはできない。そこで、この手法を拡張してPLAのネットワークを取り扱える故障シミュレーションの手法を示す。また、この手法を用いて開発した故障シミュレータPLANCS(PLA Network Crosspoint Fault Simulator)の性能を評価する。

2. PLAの並列故障シミュレーション

2.1 単一のPLAに対する並列故障シミュレーション[1]

図1に示すように入力a,b,cに1,0,0を加えたとき、積項線p1~p5はそれぞれ0,1,0,1,0となり、出力線f0~f3は1,1,0,0となる。

この入力の下でf0~f3に誤りを生じるクロスポイント故障は、以下の手順で求められる。尚、以下では積項線piと出力線fjとのクロスポイント故障を(pi, fj)、積項線piとビット線xjとのクロスポイントの故障を(pi, xj)と表わすことにする。

(1)各積項線の値とORアレイの接続情報より検出可能なORアレイ上のクロスポイント故障を次のように求める。

a) 値1の積項線とは接続のない各出力線と値1の積項線とのクロスポイント故障を全て検出可能とする。

例では (p2, f2), (p4, f2), (p2, f3), (p4, f3)。

b) 値1の積項線の1本とだけ接続している各出力線とその積項線とのクロスポイント故障を検出可能とする。

例では (p4, f1)。

c) 上記以外のクロスポイント故障は全て検出不能である。

図2に、求められたORアレイ上のクロスポイント故障を示す(網かけ部分)。

(2)各出力線に対して活性化された積項線を次のように求める。

a) 値1の積項線とは接続のない各出力線と接続しているすべての積項線。

例では f2:p1,p5 f3:p3,p5。

b) 値1の積項線の1本とだけ接続している出力線があれば、その積項線。

例では f1:p4。

c) 上記以外は活性化されていない積項線である。

図3の行列Mは上記の結果を示すものであり、値1が活性化を表わす。

(3)各ビット線の値、ANDアレイの接続情報及び(2)で求めた行列Mより、検出可能なANDアレイ上のクロスポイント故障を次のように求める。

尚、Mのある列ベクトルが0のとき、対応する積項線上のすべてのクロスポイント故障が検出不能であるので、以下の演算の対象とはならない。

a) 値1の積項線と値0のビット線とのクロスポイントの故障を全て検出可能とする。

例では (p4, a0), (p4, b1), (p4, c1)。

b) 値0のビット線の1本とのみ接続している積項線とそのビット線とのクロスポイントを検出可能とする。

例では (p1, b1), (p5, b1)。

c) 上記以外のクロスポイント故障は全て検出不能である。

図4は求められたANDアレイ上のクロスポイント故障である(網かけ部分)。

2.2 相互に接続されたPLAに対する並列故障シミュレーション

相互に接続されたPLAでは、2.1によって求めた各PLAブロックのクロスポイント故障の中で、他のPLAブロックの影響によって外部出力に誤りが伝播しないものがある。これらを除去する必要がある。

図5の回路は2.1で用いたPLA-1の後段にPLA-2とPLA-3を接続したものである。この回路に図1と同じ入力1,0,0を加えると、PLA-1の出力f0~f3は1,1,0,0となり、外部出力g0~g2は1,1,1となる。こ

の例を用いて、処理の概要を述べる。

(1)OR アレイ上の削除すべきクロスポイント故障を次の手順で求める。

まず各出力線上に、2.1で求めたクロスポイント故障が存在するかを調べる。もし存在すればその出力線に誤りを仮定する。この誤りによる信号変化に注目し、入力値の変化するPLAブロックの論理演算を一つずつ順番に行う。外部出力に変化が到達しなければ、誤りを仮定した出力線上のクロスポイント故障をまとめて削除する。

この例では、図2に示す故障(p2, f3)と(p4, f3)により、f3に1誤りが生じる。しかし、f3を入力とするPLA-3の出力はこの1誤りが生じても正常値と同じである。したがって、(p2, f3)と(p4, f3)は削除すべきクロスポイント故障である。更にf1とf2についても同様にして、(p4, f1)を削除する。

(2)AND アレイ上の削除すべきクロスポイント故障を次の手順で求める。

まず各積項線上に、2.1で求めたクロスポイント故障が存在するかを調べる。もし存在すればその積項線に誤りを仮定する。次に行列Mを参照して、この誤りが伝播する出力線を求める。(1)と同様に信号変化に注目し、入力値の変化するPLAブロックの論理演算を一つずつ順番に行う。外部出力に変化が到達しなければ、誤りを仮定した積項線上のクロスポイント故障をまとめて削除する。

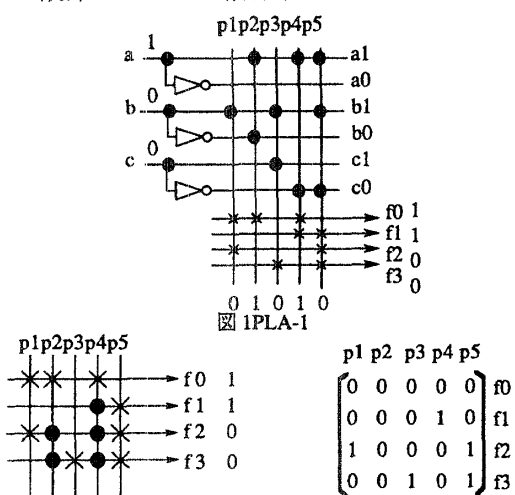


図2 OR アレイ上の検出可能なクロスポイント故障

この例では、図4の故障(p4, a0), (p4, b1), (p4, c1)によりp4に0誤りが生じる。図3の行列Mより、この誤りはf1に伝播する。f1を入力とするPLA-2の出力は、f1に誤りが生じても正常値となる。したがって(p4, a0), (p4, b1), (p4, c1)をまとめて削除できる。一方、故障(p1, b1)はp1に、(p5, b1)はp5に誤りを生じさせる。p1の誤りにより外部出力は1,1,0に、p5の誤りにより1,0,0となり、どちらも正常値と異なる。したがって、この場合は削除できない。

3. 性能評価

表1は、ベンチマークPLA*に10万個のランダムパターンを加え、PLANCSと文献[1]の故障シミュレータとの処理速度を比較した結果である。尚、評価にはSparcStation5を用いたので、文献[1]のSun3/50との処理速度比を約50倍として換算した。

表2は、元のPLA及び並直並に分解されたPLAに10万個のランダムパターンを加え、分解前と、分解後のシミュレーション時間を比較した結果である。PLANCSでは分解することにより、分解前の数倍程度の処理時間を要する。

4. むすび

相互に接続されたPLAに対する故障シミュレーション手法を提案した。またその手法にもとづく故障シミュレータPLANCSを開発し、その有効性を確認した。

参考文献

- [1]Eli Chiprout, Janusz Rajski and Markus Robinson, "Parallel PLA Fault Simulation Based on Boolean Vector Operation", Proc. of ICCAD, pp.194-197,1988.
- [2]Ruey-Sing Wei and Alberto Sangiovanni-Vincentelli, "PLATYPUS: A PLA Test Pattern Generation Tool", Proc. of 22nd DAC, pp.197-203,1985.
- [3]Hongtao P.Chang and Jacob A.Abraham, "Use of High Level Descriptions for Speedup of Fault Simulation", Proc. of ITC'87, pp.278-285,1987.

*ESPRESSO-II添付, Leuven University, 九州工業大学 笹尾勲教授, Brayton教授 提供のものを用いた。

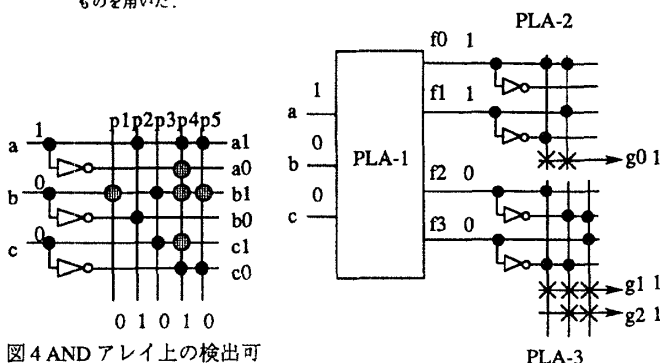


図4 AND アレイ上の検出可能なクロスポイント故障

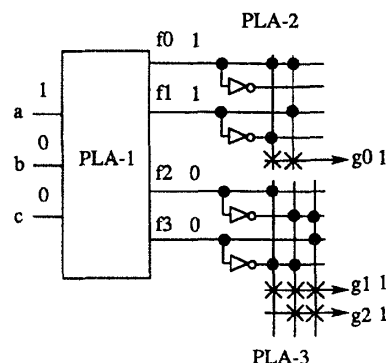


図5 PLA ネットワークの例

表1 文献[1]とPLANCSの処理時間の比較

PLA	入力数	積項線数	出力数	PLANETS(秒)	文献[1](秒)
ti	14	213	67	76.9	117.9
tial	14	579	8	98.3	235.4
bc0	26	178	11	41.7	53.6
x2dn	82	104	56	105.5	85.4

表2 PLANCSの処理時間

PLA	分解前		分解後			分解後/分解前 時間比
	クロス ポイント数	時間(秒)	ブロック 数	総クロス ポイント数	時間(秒)	
vg2	6,380	20.2	8	780	49.6	2.5
b2	5,194	29.2	6	9,159	100.3	3.4
kawadafa	18,093	49.3	8	2,265	131.5	2.7