

高速故障シミュレータを用いたテスト生成

7U-2

伊藤雅樹 石浦菜岐佐 矢島脩三
(京都大学工学部)

1. あらまし

ベクトル計算機上に実現された高速故障シミュレータを用いた、ランダムパターンによるテスト生成手法を提案する。故障及びその伝搬経路を活性化する要求を入力信号値確率(外部入力論理値1となる確率)に反映させ、確率に従うランダムパターンを発生し、ベクトル計算機で実行することにより、高速なテスト生成が可能になる。なお、対象は組合せ回路であり、故障は単一縮退故障を仮定する。

2. 動的二次元並列故障シミュレータ

テスト生成において故障シミュレーションは繰り返し実行されるため、その高速化の要求が強い。この要求に対して我々はベクトル計算機を用いることを提案した^[1]。ベクトル計算機はベクトルに対する様々な演算を超高速に処理するが、その性能を十分に引き出すためにはベクトル計算機を指向したアルゴリズムが必要である。

ベクトル計算機用のアルゴリズムとして、我々は動的二次元並列法を考案した。動的二次元並列法では、1)入力パターンと故障の両方向に並列化することにより大きなベクトル長を得、2)二つの並列要素の大きさをバスごとに変化させることにより故障のドロップを効果的に行い、3)選択的追跡を行い、無駄なゲート評価を除いている。ベクトル計算機FACOM VP-200上での性能評価では、従来に比較し10~15倍高速であるという結果が得られ、特に大量のパターンに対するシミュレーションに有効であった。

3. テスト生成手法

本稿では、前章の高速故障シミュレータを用いてテスト生成を高速に行うことを提案する。大量のパターンの処理に対して有効であるという特徴を活かすために、ランダムパターンから故障シミュレータによりテストパターンを選ぶ。しかし、純粋なランダムパターンでは高い検出率を得るためには膨大な数をシミュレートしなければならない。そこで、故障を検出する可能性の大きいランダムパターンを発生することを考える。そのためには、非一様な入力信号値確率に従うパターンを発生することが有効である^[2]。

我々は効率の良い入力信号値確率の計算法を新し

く考案した。入力信号値確率の計算は故障が検出されるたびに全ての未検出故障を対象に行われるべきであり、従来の複雑な計算ではその実効性に問題があった。そこで、高速な計算が可能で、しかも未検出故障の活性化の要求、及びその伝搬経路の活性化の要求を直接反映する計算法を提案する。また、確率に従うランダムパターンは、複数の線形帰還シフトレジスタの出力に論理演算を適用することによって、高速に発生する。

3.1 入力信号値確率の計算

入力信号値確率に未検出故障とその伝搬経路の活性化の要求を反映させるために、各信号線に対して0要求カウンタ(RC0)と1要求カウンタ(RC1)を用意する。それぞれ、故障及び伝搬経路の活性化のためにその信号線の信号値を0又は1にする要求の数を表す。次に外部出力側から外部入力側へ逆レベル順に要求カウンタの値を伝搬させ、外部入力の要求カウンタの値が求まったとき、 $RC1/(RC0+RC1)$ を入力信号値確率とみなすのである。手続きの詳細を以下に示す。

Step 1. 故障の存在する信号線において、s-a-0故障のときはRC1を、s-a-1故障のときはRC0を1増やす。これを全ての未検出故障に対して行う。

Step 2. 各未検出故障の全伝搬経路上のゲートの伝搬経路以外の入力線の要求カウンタの値を、伝搬経路を活性化するように増加する。

Step 3. 外部出力側から外部入力側へ、逆レベル順に要求カウンタの値を伝搬させる。ゲート出力の要求カウンタ値は、ゲートの種類に応じた関数によって変換され、ゲート入力の要求カウンタの値に加えられる(図1)。ファンアウトでは、全ての枝の要求カウンタの値の和が、ファンアウトの茎の要求カウンタに加えられる(図2)。

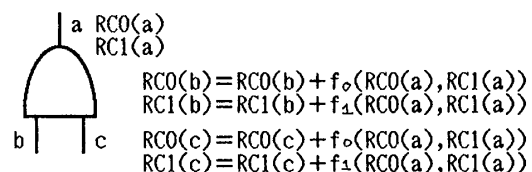


図1. 要求カウンタの値のゲートにおける伝搬

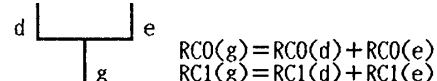


図2. 要求カウンタの値のファンアウトにおける伝搬

表1. 実験結果

回路名	本稿の手法					全て0.5の入力信号値確率のランダムボタン				
	シミュレートしたランダムボタン数	選ばれたテストボタン数	未検出故障数	検出率 (%)	CPU時間 (秒)	シミュレートしたランダムボタン数	選ばれたテストボタン数	未検出故障数	検出率 (%)	CPU時間 (秒)
C432	1184	130	4	99.2	.152	4192	112	←	←	.166
C499	2528	145	8	98.9	.207	8160	105	←	←	.211
C880	736	136	0	100.0	.317	18400	179	←	←	.446
C1355	10208	162	8	99.5	.398	10720	156	←	←	.377
C1908	120288	258	9	99.5	.888	22496	248	←	←	.584
C2670	524288	211	150	94.5	12.9	←	204	207	92.5	13.4
C3540	65504	402	137	96.0	1.80	←	383	←	←	1.67
C5315	36832	374	59	98.9	3.49	16352	321	←	←	3.14
C6288	256	46	34	99.6	.949	1024	32	←	←	1.02
C7552	524288	483	222	97.1	16.8	←	507	228	97.0	15.8

“←”は本稿の手法の対応する欄の値と同じ値であることを示す

Step 4. 外部入力全ての要求カウンタの値が求まると、 $RC1/(RC0+RC1)$ を各外部入力の信号値確率をみなす。ただし $RC0+RC1=0$ のときは信号値確率は0.5とする。

上記の手続きによって計算される確率は再収斂のため、故障および伝搬経路の活性化の要求を正確に反映したものとはならない。なおStep 3における関数としては、例えば n 入力ANDゲートの場合は $(RC0+RC1) \cdot \sqrt[n]{x}$ を用いている。ただし x はゲート出力の $RC1/(RC0+RC1)$ である。

3.2 確率に従うランダムボタンの発生手法

入力信号値確率が求まると、それらに従うランダムボタンを発生する。我々は複数の一様なランダムボタンに論理演算を施すことによってこれを実現した。例えば、確率0.25で論理値1を得るには確率0.5で発生する二つの論理値1のANDをとる。確率0.5で発生する論理値1は、線形帰還シフトレジスタの出力として高速に得ることができ、ビットワイス論理演算により、32ボタンを同時に発生できる。

この方法はビットワイス論理演算及びテーブル参照によって実現できるため、確率に従うランダムボタンを高速に発生でき、またベクトル処理や我々の高速故障シミュレータとの整合性がよい。

4. 実験結果

前節までに述べた手法を用いた自動テスト生成システムをFACOM VP-400E上にFortran77で実現し、性能評価を10個のベンチマーク回路^[3]を用いて行った。テスト生成は以下の三つの条件のうち一つが成り立った時点で打ち切った。

- 1) 目標検出率(100%)が得られた。

- 2) 予定のランダムボタン数(512K)をシミュレートした。

- 3) ある時点まででシミュレートしたボタン数のうち最後の60%のボタンをシミュレートする間に故障が一つも検出されなかった。

表1に結果を示す。比較のために確率0.5で発生したランダムボタンによるテスト生成の結果も示す。表から次のことが読み取れる。

- 1) C2670とC7552を除いて数秒以下でテスト生成が終了している。C2670とC7552でも20秒以下である。選ばれたテストボタン数は比較的多いが、必要とするCPU時間は極めて少ない。
- 2) C1908とC5315を除いて同じ検出率を得るのに必要なランダムボタン数は本稿の手法の方が少ない。ランダムボタンでは高い検出率を得難い回路として知られるC2670とC7552では検出率が向上している。

5. おわりに

ベクトル計算機を用いた高速なテスト生成手法を提案した。入力信号値確率の計算法として、単純な方法を採用したが、そのベクトル化やさらに有効なテスト生成を可能にする計算法の開発が今後の課題である。

謝辞：御討論頂いた矢島研究室の皆様へ感謝します。

参考文献

- [1]N.Ishiura他：ICCAD-87,(Nov. 1987).
- [2]R.Lisanke他：ICCAD-86,(Nov. 1986).
- [3]F.Brglez他：ISCAS85,(June 1985).