ベクトル計算機による高速故障シミュレーションのための動的二次元並列法†

石浦 菜岐佐†† 伊藤 雅樹†† 矢島 脩三†††

故障シミュレーションの高速化の新しいアプローチとして，ベクトル計算機を利用する方法を提案する．ゲート・レベルの組合せ回路を対象とする，2値の零遅延シミュレーションについて，並列法を基本とする計算手法を開発した．並列法は処理単位を1ワードから複数ワードに拡張することにより，処理ベクトル長が十分長い場合には，ベクトル演算の利用により最大で20倍以上加速できる（計算機は FACOM VP-200）．しかし，故障シミュレーションを実際に検査入力の生成や評価に利用する場合には，故障のドロップ（検出された故障を以後の処理の対象からはずすこと）が行われるため，単純な並列法では十分な処理ベクトル長が得られない，あるいは処理ベクトル長を長くとろうとすると計算量が増加するという問題が生じる．これらの問題に対処するため我々は，1) 故障，入力の両方向に並列化を行い，2) 故障，入力の並列度をパスごとに動的に変化させることにより，故障のドロップに対応しつつ処理ベクトル長を確保する，動的二次元並列法を新たに提案する．本論文では，さらに計算の無駄を削減するため，選択的追跡を導入している．これは，単一故障伝播を拡張した複数故障伝播の概念に基づき，無駄な計算量を削減しつつ効率の良いベクトル処理を実現するものである．

1. はじめに

故障シミュレーションは，故障を仮定した論理回路の動作をシミュレーションするものであり，論理回路に対する故障検査入力の生成・評価に不可欠なものとなっている[1]．検査入力の生成において必要となる故障シミュレーションの計算量は，回路を構成するゲート数 n に対し $O(n^2)$〜$O(n^3)$ と言われており[2]，回路規模の増大に伴い膨大な計算時間が必要となる．これに対し，アルゴリズムの改良による故障シミュレーションの高速化[3),4),10)]，故障シミュレーションにかわる検査入力の評価手法の開発[5]，等の研究が行われている．これに対し，本論文ではベクトル計算機の利用により，故障シミュレーションを高速化する方法を提案する．

ベクトル計算機は演算パイプライン方式のスーパ・コンピュータであり，ベクトル・データ（配列データ）に対する同一の繰り返し処理を演算パイプラインで実行することにより，計算の高速化を実現する．その最大性能は数百 MFLOPS であり，超大型汎用機の数十倍にも達する[13),14)]．しかしこの性能を引き出すためには，プログラムの大部分がベクトル命令で実行でき，かつ処理ベクトル長が十分長いことが不可欠であり，アルゴリズム，コーディングの両面にわたって処理のベクトル化を考えなければならない．

本論文ではゲート・レベルの組合せ回路を対象とする，2値の零遅延シミュレーションについて考える．故障シミュレーションの主な計算手法には，1) 並列法，2) 演繹法，3) 同時法が知られている[3]．ゲート数が大きい場合には，一般に並列法よりも演繹法，同時法のほうが計算量が少なくてすむことが知られており[1]，最近では同時法を用いるのが主流となりつつある．しかし我々は以下の理由から，並列法を基本としたベクトル計算機向きの計算法を開発している．

1) ベクトル計算機による加速：並列法のもつデータ構造，単純な処理過程は同時法よりもベクトル計算機になじみやすい．論理シミュレーションの場合においても，イベント法[3]よりもコンパイラ法[3]に基づくもののほうが，ベクトル処理による加速率が大きいことが示されている[6]．

2) 並列法の改良の可能性：並列法も PPSFP 法[10]のような工夫を加えることによって，計算量のオーダを同時法に近づけ得る可能性があり，ゲート数が大きな場合においても同時法をしのぐことがあり得る．

並列法は計算機のビットワイズ論理演算命令を利用して，複数のゲート評価を並列に実行する方式であり，並列化の方法によって，故障並列法と入力並列法に分類される[12]．我々は各々の方法について処理単位を1ワードから複数ワードに拡張する拡張並列法により，処理ベクトル長が十分長い場合には，ベクトル演算の利用により最大で 20 倍以上の高速化が達成され

† Dynamic Two-Dimensional Parallel Simulation Technique for High-Speed Fault Simulation on a Vector Processor by NAGISA ISHIURA, MASAKI ITO and SHUZO YAJIMA (Department of Information Science, Faculty of Engineering, Kyoto University).

†† 京都大学工学部情報工学教室

ることを示した[8]（計算機は FACOM VP-200）．故障シミュレーションを実際に検査入力の生成や評価に利用する場合には，故障のドロップ（検出された故障を以後の処理の対象からはずすこと）が行われる．このため，単純な拡張並列法では十分な処理ベクトル長が得られない，あるいは処理ベクトル長を長くとろうとすると計算量が増加するという問題が生じる．これらの問題に対処するため我々は，

1) 故障，入力の両方向に並列化を行う，

2) 故障，入力の並列度を動的に変化させる

ことにより，故障のドロップを考慮しつつ処理ベクトル長を確保する，動的二次元並列法を新たに提案する．本論文の手法では，さらに計算の無駄を削減するため，選択的追跡を導入している．これは，単一故障伝播[10]を拡張した複数故障伝播の概念に基づき，無駄な計算量を削減しつつ効率の良いベクトル処理を実現するものである．

以下2章では，動的二次元並列法の概念について述べた後，3章で複数故障伝播による選択的追跡の実現法について述べる．4章では試作したシミュレータを用いた実験の結果に基づき，性能評価を行う．

2. 動的二次元並列法

2.1 故障シミュレーション

故障シミュレーションは，1) 回路情報（ゲートの種類および結線の記述），2) 回路に起こり得る故障の集合，3) 回路に対する入力パタンの集合，を入力として，各故障を設定したときの各入力パタンに対する回路の出力を計算するものであり，次のような応用に用いられる．

1) 与えられた入力パタン集合で検出できる故障，あるいは検出できない故障を求める．

2) 与えられた入力パタン集合の検出率を求める．

3) 与えられた入力パタン集合のなかから有効なものだけを選んで，検査入力組をつくる．

4) 故障辞書を作成する．

本論文では，スキャン・デザイン方式[7]により設計された回路を対象とし，ランダム・パタン等の大量の入力パタン集合に対して 1)～3) を行うための故障シミュレーションを興味の対象としている．このような応用では，故障シミュレーションを組合せ回路の2値，零遅延のものに制限することで，極めて効率の高い処理が実現できる[10]．また，与えられた入力パタン集合に対して故障が検出されるか否かのみを問題にする場合には，ある入力パタンで検出されることが判明した故障は，未検出故障のリストから削除し，他の入力パタンでのシミュレーションの対象からはずしてもさしつかえない．これにより処理量の削減を図ることを故障のドロップという．なお本論文では故障の仮定としては単一縮退故障を考えている．

2.2 二次元並列法

並列法は計算機のビットワイズ論理演算命令を用いて計算の高速化を図るもので，1ワードが b ビットならば，一命令で b 回のゲート評価を一度に行う方法である[3]．並列法は更に故障並列法と入力並列法に分類される[12]．故障並列法は1ビットに1つの故障ケースを割り当て，（1つの入力パタンについて）一度に b 個の故障をシミュレーションするものである．これに対し，入力並列法は1ビットに1つの入力パタンを割り当て，（1つの故障について）b 個の入力パタンを一度にシミュレーションする方式である．

いずれの方法も，処理単位を1ワードから w ワードに拡張し，一度に $b \times w$ 個の故障ケースあるいは入力パタンをシミュレーションするようにできる．このように，故障並列法，入力並列法を拡張した計算方式を，それぞれ拡張故障並列法，拡張入力並列法と呼ぶことにする．処理単位を複数ワードに拡張することにより，一般のスカラ計算機上でも論理演算の速度が向上することが知られているが[4]，ベクトル計算機上ではベクトル論理演算命令の利用により，格段の速度向上が実現できる．**図1**は拡張並列法におけるベクトル長と加速率の関係を示したものである[8]．計算機は FACOM VP-200 であり，ここでの1ワードは 32 ビットである．ベクトル長が十分長ければ（700 以上），スカラ実行に比べ 20 倍以上の高速化が実現される．

拡張入力並列法における処理ベクトル長は［入力パタン数］/b で与えられ，入力パタン長を変えることに

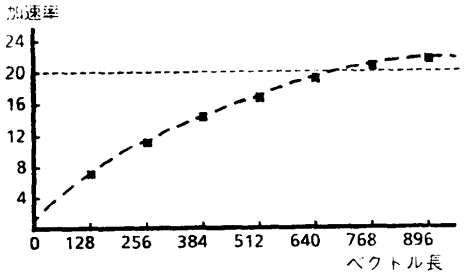

図1 拡張入力並列法による加速率
Fig. 1 Acceleration ratio in the extended pattern parallel simulation.

より，任意に変えることができる．これに対し拡張故障並列法では，処理ベクトル長は［故障数］/b であり，回路に起こり得る故障の数で制限されてしまうため，十分な処理ベクトル長が得られない場合がある．このような場合には，故障並列法，入力並列法の両者を組み合わせた二次元並列法が有効となる．二次元並列法は具体的には，故障並列法において，多数のパタンを一度にシミュレーションする，あるいは入力並列法において多数の故障ケースについて一度にシミュレーションするものである．以下では一度に処理する故障数 f を故障並列度，入力パタン数 p を入力並列度と呼ぶことにする．

2.3 並列度の動的な決定

拡張並列法や二次元並列法によれば，故障シミュレーションは 20 倍以上高速化され得るが，これはあくまで十分な処理ベクトル長が得られた場合の話である．実際に，故障シミュレーションを検査入力の生成・評価へ応用する場合，故障のドロップを行うため，入力並列度，故障並列度は制限される．**図 2** はシミュレーションの進む様子を示している．各々の長方形の面積は各パス（ひとまとまりの入力パタンの読み込みから，正常時および故障設定時のシミュレーションまでをパスという）におけるベクトル長，あるいは計算量に対応する．図 2 (a) に示すように，シミュレーションの初期には多くの未検出故障が存在し，大きなベクトル長が得られるが，シミュレーションが進むにつれて未検出故障の数は減少するため，最後には故障並列度が極めて小さくなり，十分なベクトル長が得られない可能性がある．しかし，後半に大きなベクトル長を確保するために，長いパタンでシミュレーションを行うと，図 2 (b) に示すように，前半のパスにおいて大幅な計算量増加につながる．このように，故障のドロップを考慮した場合，単純な二次元並列法では，処理ベクトル長が十分とれない，あるいは処理ベクトル長を十分とろうとすれば計算量が増加するという問題が生じ，処理時間の短縮を図ることが難しくなる．このような問題に対処するため，我々は故障並列度 f と入力並列度 p を固定とせず，パスごとに適当な値に変えることを提案する．すなわち，図 2 (c) に示すように，初期のパスでは故障並列度を大きく入力並列度を小さくとり，後期のパスでは故障並列度を小さく入力並列度を大きくとるのである．このように，二次元並列法において，二つの並列度を状況に応じて変化させることにより，故障のドロップを考慮しつつ，十分な処理ベクトル長を確保する手法を，動的二次元並列法と呼ぶ．

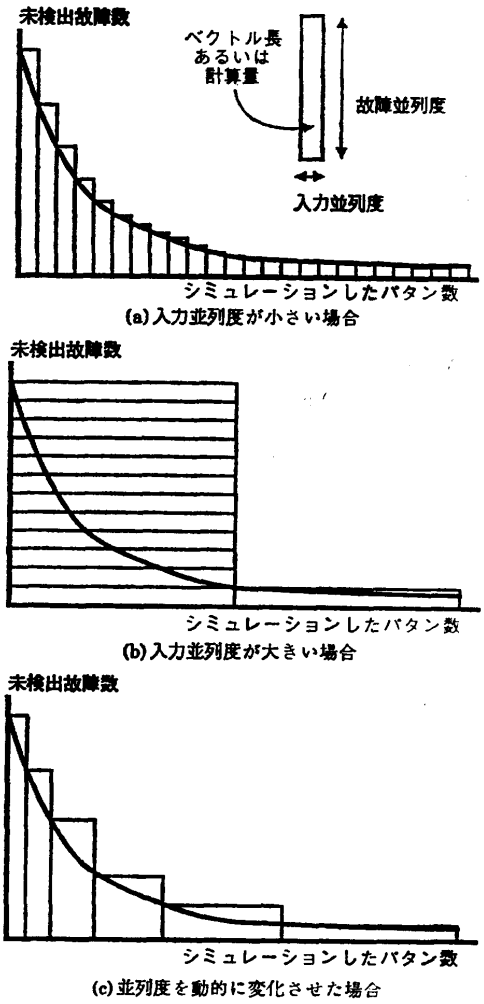

図 2 故障のドロップを行う故障シミュレーション
Fig. 2 Fault simulation with fault dropping.

3. 複数故障伝播

3.1 選択的追跡

選択的追跡は，信号値が前回の入力パタンによる値や正常値と異なる場合のみを処理の対象とすることにより，計算量の削減を図る手法である．並列法と選択的追跡を組み合わせる方法はいくつか考えられるが，本論文では PPSFP 法[10]と同様に，

1) まず，故障を設定しない回路のシミュレーションを行い，全信号線についてその信号値を記憶する．

2) 次に，仮定された故障について，故障が設定された点から外部出力に向かって故障の影響を伝播することにより，故障時の回路状態を計算する．

という方法をとる．PPSFP 法では，2) において入力並列法により故障を一つずつ処理する（単一故障伝播）が，我々は動的二次元並列法により複数の故障を一度に処理する．このような処理方式を単一故障伝播に対して複数故障伝播と呼ぶことにする．複数故障伝播による故障シミュレーションの手続きの詳細を図 3 に示す．⑤では④で得た p パタンについて，正常時の回路のシミュレーションを行い，全ゲートについて信号値を記録する．⑦ではこの p パタンに対し⑥で選んだ f 故障について動的二次元並列法による故障シミュレーションを行い，故障の影響を出力側へ伝播する．

3.2 選択的追跡の実現手法

3.2.1 故障シミュレーションにおける選択的追跡

本論文では選択的追跡を，追跡領域の限定，および追跡打ち切りという概念でとらえる．追跡領域の限定とは，回路の結線情報を用いて，ゲート評価の対象を，信号値変化の影響が及ぶ可能性のあるゲートに限定することを意味する．また，追跡打ち切りは，変化の影響が消滅したと判断できれば，以後の追跡を打ち切ることを意味する．

3.2.2 追跡領域の限定

回路中のある故障の影響を受ける可能性のあるゲートは，その故障から外部出力に至るパス上のゲートに限られる．このようなゲートの集合を，その故障の影響領域ということにする．追跡領域の限定は，ゲート評価の対象を，現在設定されている故障の影響領域に限定することである．追跡領域の限定は，レベル・マッピング・イベント法により実現する．すなわち，各レベルごとに，評価ゲート・リストを用意し，初期設定として，故障が設定されたゲートを対応するレベルの評価ゲート・リストに登録しておく．故障の伝播は，レベル番号の若い順に評価ゲート・リストからゲートを取り出しては評価し，そのファンアウト先のゲートを対応するレベルの評価ゲート・リストに登録する，という手順を繰り返すことにより実現される．

複数故障伝播において設定された複数の故障の影響領域がすべて一致するときには，ゲート評価ののべ回数はこれらの故障を個々に処理する場合と同じであるが，設定された故障の影響領域の共通部分が小さいときには計算の無駄が多くなる．したがって，図3の⑥において f 個の故障を選ぶ際には，選んだ故障の影響領域がなるべく重複していることが望ましい．我々は，シミュレーション実行時のオーバヘッドを避けるために，回路データを作成する前処理の段階で，未検出故障リスト中になるべく近傍の故障が並ぶように配置を行い，⑥では単純に未検出故障リストの先頭から必要な個数だけ順次故障を取り出すという方式を採っている．

3.2.3 追跡打ち切り

故障を一つずつ処理する場合の追跡打ち切りは，ある信号線のある故障時の信号値が，現在シミュレーションしているすべての入力パタンについて，正常時の信号値と一致しているとき，故障の伝播を打ち切ることである．複数の故障を一度に処理する場合には，これがさらに次の２つの概念に分けられる．

1) 経路追跡打ち切り：ある信号線の信号値が，全入力パタン，全故障について，正常時の値と一致したときその伝播経路に対する追跡を打ち切る．

2) 故障追跡打ち切り：ある故障に対し，その影響の及ぶ全伝播経路について，全入力パタンの値が，正常時の値と一致したときその故障に対する故障伝播を打ち切る．

経路追跡の打ち切りは，ゲートの出力値が現在設定

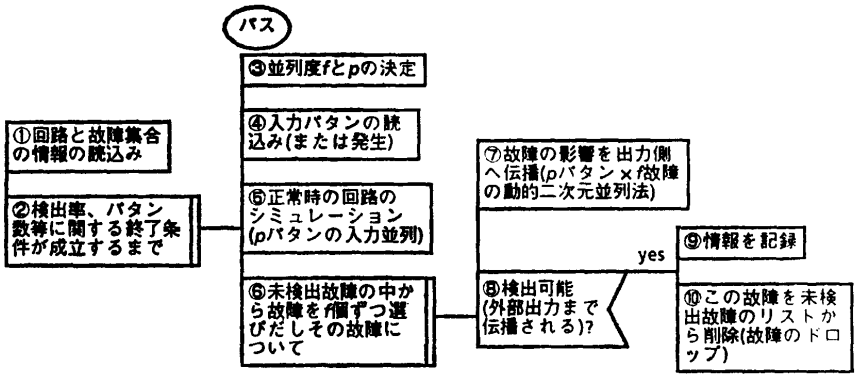

図 3 複数故障伝播による故障シミュレーションの手続き
Fig. 3 Procedure of the fault simulation based on the multiple fault-propagation.

しているすべての故障について正常時の値と一致する場合に，ファンアウト先のゲートの評価ゲート・リストへの登録を中止することで容易に実現できる．故障追跡の打ち切りは，現在設定している故障の影響が全伝播経路で消滅したことを判定し，消滅していればその故障を処理対象から除外するという操作により実現する．故障の影響の消滅は，故障ごとにその故障の影響が及んでいるゲートのレベルの最大値を記憶しておき，現在ゲート評価をしているレベルの番号との大小比較により判定するという方式をとっている．

3.3 並列度の決定法

動的二次元並列法では，図3の③における故障並列度 f と入力並列度 p の決定が重要となる．考慮すべき要素を以下に示す．

1) 故障のドロップの効果を生かすという観点からは，p は小さいほうが良い．

2) 正常時の信号値を記憶する記憶量の観点からは，p は小さいほうが良い．

3) 追跡打ち切りの条件は p が小さいほど成立しやすい．

4) 正常時のシミュレーション等はベクトル長が p なので，p は大きいほうが処理効率が良い．

5) 追跡領域の限定を行う場合，f が小さいほど無駄な処理が増えない．

これらの条件を考慮して，処理時間最適な p を決定できればよいのであるが，条件を完全に定量化することは難しいので，1パスごとに未検出故障の減少率 d をみて，次のパスの入力並列度 p_{next} を現パス入力並列度 p の2倍，1倍，1/2倍のいずれかにするという方式を使っている．すなわち，

d＝(現パス終了時の未検出故障数)
　　/(現パス開始前の未検出故障数)

であり，

$d_2<d$ のとき，$p_{next}=\min(2p, 32\ maxp)$

$d_1\leqq d\leqq d_2$ のとき，$p_{next}=p$

$d<d_1$ のとき，$p_{next}=\max(p/2, 1)$

$f=32maxfp/p$

とする．ただし，$maxp$ は入力並列度の最大値［ワード］，$maxfp$ は動的二次元並列法におけるベクトル長の最大値［ワード］である．d_1, d_2 は，$r=0.1$ として，

$$d_1=(rp/(rp+64))^2,\ d_2=(rp/(rp+32))$$

を用いている．この式は，故障が p パタンごとに d 倍に減少していくという仮定から求めた故障シミュレーションの計算量 C と，ベクトル長が P のときの1演算あたりの処理時間 T ($T=\alpha P+\beta$ で近似) の積 CT が小さくなるように定めたものである．

4. 性能評価

4.1 処理速度

前章までに述べたシミュレーション方式に基づくシミュレータを Fortran 77 で作成し，その処理速度と使用記憶量に関する性能評価を行った．使用計算機は FACOM VP-200[13]（京都大学大型計算機センター）である．実験は，10種類のベンチマーク回路[9]に対し，512 K 個（16 K ワード相当）のランダム・パタンをシミュレーションするのに要する CPU 時間を測定するというものである．ベクトル化の効果を評価するため，ベクトル演算命令を用いないモード（以下スカラ実行と呼ぶ）での実行も行った．また，選択的追跡の効果を評価するため，選択的追跡を行わないもの，追跡領域の限定のみ行うものについても測定を行った．選択的追跡のバージョン，実行モードを**表1**にまとめる．

実験結果は**表2**のとおりである．最大処理ベクトル長 $maxfp$ はいずれも 1024 である．最大入力並列度 $maxp$ については，Vector 以外は 256 としたが，選択的追跡を行わない Vector は，同一の記憶量で $maxp$ を大きくとれるため，その値を 1024 とした．表より以下が観測される．

1) ベクトル実行の効果

Scalar^{++} と Vector^{++} の比較より，ベクトル実行による加速率は数倍から15倍程度であることがわかる．2章で述べた単純な拡張並列法の最大加速率（20倍以上）には満たないものの，大規模な回路では十分にベクトル計算機の性能を引き出しているといえる．

2) 選択的追跡の効果

Vector と Vector^{+} の比較より追跡領域の限定の効果が，Vector^{+} と Vector^{++} の比較より追跡打ち切りの効果がわかる．どちらがどの程度有効かは回路によ

表1 選択的追跡のバージョンと実行モード
Table 1 Version of the selective tracing and execution mode.

	Vector	Vector^{+}	Vector^{++}	Scalar^{++}
追跡領域の限定	—	○	○	○
追跡打ち切り	—	—	○	○
実行モード	ベクトル実行	ベクトル実行	ベクトル実行	スカラ実行

表 2 512K 個のランダム・パタンのシミュレーションの結果
Table 2 Result of the simulation for 512K random patterns.

Circuit	Number of gates	Number of faults	Final coverage [%]	Number of untested faults	Simulation CPU [sec]			
					Vector	Vector^{+}	Vector^{++}	Scalar^{++}
C 432	203	524	99.24	4	.176	.280	.275	2.560
C 499	275	758	98.94	8	.461	.791	.571	6.236
C 880	469	942	100.0	0	.179*	.462	.490	4.651
C1355	619	1574	99.49	8	.930	1.355	.871	10.979
C1908	938	1879	99.52	9	1.930	1.381	1.387	16.026
C2670	1566	2447	91.37	237	57.695	10.318	8.330	99.474
C3540	1741	3428	96.00	137	19.037	15.440	3.935	48.463
C5315	2608	5350	98.89	59	17.864	6.962	3.489	41.818
C6288	2480	7744	99.56	34	15.605	22.388	2.604	38.194
C7522	3827	7550	96.89	235	114.868	19.939	11.518	154.885

* 全故障検出後のシミュレーションを打ち切った

り異なるが，一方があまり効果を持たないときには他方が有効であるという，相補的効果が観測される．全体として，選択的追跡の導入により若干性能が低下した回路もみられるが，大きな計算量を要していた回路に対しては，10倍程度の性能改善が達成されている．

本論文の実験結果は，文献 10), 11) の結果（同一のベンチマーク回路に対して同様の実験が行われている）と比較しても非常に高速であり，十分実用的なレベルにあるといえる．また，本論文のシミュレータは，最近国内で開発されたベクトル計算機[13),14)]に共通な基本的ベクトル命令のみを用いて実行できるため，容易に移植が可能である．我々は VP-200 と併せて HITAC S-810/20（東京大学大型計算機センター）上でも同じ実験を行ったが，プログラムを変更することなくベクトル化可能で，VP-200 の場合とほぼ同じ性能を得ることができた．

4.2 必要記憶量

必要記憶量は回路，故障の情報のためのものと，信号線の信号値パタンのためのものがある．前者は，通常の表駆動方式の並列シミュレータと同程度であるが，後者は図3の⑤と⑦の処理において，ベクトル長の増大とともに大きな量となる．⑤では⑦で選択的追跡を行うために，すべてのゲートについてシミュレーション結果である正常出力を記憶しておく必要がある．回路のゲート数を n とすると，必要記憶量は $n\times maxp$ [ワード] である．⑦では p パタンについて f 個の故障に対するシミュレーションを行うため，1ゲートあたり $p\times f/32$ [ワード] のベクトルが必要となる．しかし，⑦ではすべてのゲートについてその信号値を記憶しておく必要はなく，文献 6) のように信号値を格納する領域を再利用する手法により，記憶量を削減できる．故障伝播のある時点で記憶しておくべき信号値ベクトルの最大数を w とすると，必要記憶量は $w\times maxfp$ [ワード] となる．今回の実験で最大の回路に要した記憶量は約 10 MB であった．

5. むすび

ベクトル計算機向きの故障シミュレーション手法として，動的二次元並列法を提案した．複数故障伝播の概念に基づいて，追跡領域の限定，追跡の打ち切りなどの選択的追跡手法を導入し，処理効率を落とすことなくベクトル長を確保することに成功した．性能評価を行った結果，ベクトル実行により 10〜15 倍の高速化が達成され，非常に高速なシミュレーションが可能となることがわかった．このような高速な故障シミュレーション技法は，ランダム・パタンを用いた故障検査入力生成や，BIST の故障検出率の評価などに有効であると考えられる．

謝辞 本論文の研究に関して，有益な御助言をいただきました日本電気の河合正人氏に感謝いたします．また，ベンチマーク回路データの入手にご厚意をいただいた明治大学の藤原秀雄先生に感謝いたします．本論文の拡張並列/二次元並列故障シミュレータのコーディングおよび実験に御協力いただいた久米政輝氏，大西達也氏，中田秀男氏に感謝します．また，本研究の遂行にあたり，御助言，御協力いただいた本学の安浦寛人助教授に感謝します．最後に，平石裕実助教授，高木直史氏，荻野博幸氏を始め，御討論いただいた矢島研究室の諸氏に感謝します．

参考文献

1) 藤原秀雄：テスト生成と故障シミュレーション，情報処理，Vol. 25, No. 10, pp. 1119-1124 (1984).
2) 山田昭彦，船津重宏，黒部恒夫：論理回路の試験，診断，情報処理，Vol. 22, No. 8, pp. 770-777 (1981).
3) Breuer, M. A. and Friedman, A. D.: *Diagnosis & Reliable Design of Digital Systems*, p. 308, Computer Science Press, California (1976).
4) 高松雄三，樹下行三：並列故障シミュレーションにおけるワード数の効果について，第 32 回情報処理学会全国大会論文集，4 U-13, pp. 1987-1988 (Mar. 1986).
5) Jain, S. K. and Agrawal, V. D.: STAFAN: An Alternative to Fault Simulation, *Proc. of 21st D. A. Conf.*, pp. 18-23 (June 1984).
6) 石浦菜岐佐，安浦寛人，矢島脩三：ベクトル計算機による論理シミュレーションの性能評価，電子通信学会技術研究報告，CAS 86-82, pp. 25-32 (Sept. 1986).
7) Williams, T. W. and Parker, K. P.: Design for Testability—A Survey, *IEEE Trans. Comput.*, Vol. C-31, No. 1, pp. 2-15 (1982).
8) 石浦菜岐佐，久米政輝，安浦寛人，矢島脩三：ベクトル計算機による並列故障シミュレーション，電子通信学会技術研究報告，FTS 86-4, pp. 25-30 (May 1986).
9) Brglez, F. and Fujiwara, H.: A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in FORTRAN, Special Session on ATPG and Fault Simulation, *International Symposium on Circuits and Systems—ISCAS 85* (June 1985).
10) Waicukauski, J. A., Eichelberger, E. B., Forlenza, D. O., Lindbloom, E. and McCarthy, T.: Fault Simulation for Structured VLSI, *VLSI Systems Design*, pp. 20-32 (Dec. 1985).
11) Briers, A. J. and Totton, K. A. E.: Random Pattern Testability by Fast Fault Simulation, *Proc. of 1986 International Test Conference*, pp. 274-281 (Sept. 1986).
12) 樹下行三，浅田邦博，唐津 修：岩波講座マイクロエレクトロニクス 4 VLSI の設計 II, pp. 241-254, 岩波書店，東京 (1985).
13) 速さを競うスーパーコンピュータ，日経エレクトロニクス，1983 年 4 月 11 日号 (No. 314), pp. 105-184 (Apr. 1983).
14) スーパーコンピュータ SX システム，日経エレクトロニクス，1984 年 11 月 19 日号 (No. 356), pp. 237-271 (Nov. 1984).

(昭和 62 年 10 月 12 日受付)
(昭和 63 年 3 月 9 日採録)

石浦菜岐佐（正会員）

昭和 36 年生．昭和 59 年京都大学工学部情報工学科卒業．昭和 61 年同大学院修士課程修了．昭和 62 年 1 月より京都大学工学部助手．論理回路の CAD/DA の研究に従事．電子情報通信学会会員．

伊藤 雅樹（学生会員）

昭和 39 年生．昭和 62 年京都大学工学部情報工学科卒業．現在，同大学院修士課程に在学中．論理回路のテスト生成の研究に従事．

矢島 脩三（正会員）

昭和 8 年生．昭和 31 年京都大学工学部電気工学科卒業．同大学院博士課程修了．工学博士．昭和 36 年より京大工学部に勤務，昭和 46 年情報工学科教授．昭和 35 年京大第一号計算機 KDC-1 を設計稼動，以来，計算機，論理設計，オートマトン等の研究教育に従事，著書は「電子計算機の機能と構造」(岩波，57 年) 等，本学会元常務理事，元会誌編集委員 (地方)，元 JIP 編集委員．電子情報通信学会元評議員およびオートマトンと言語研専元委員長，North-Holland 出版 IPL 編集委員，IEEE Senior Member.