「並列/分散処理による LSI-CAD ワークショップ」 平成 6 年 12 月

テスト生成における並列処理の性能評価†

藤井 貴晴　　井上 智生　　藤原 秀雄

奈良先端科学技術大学院大学 情報科学研究科

あらまし　論理回路に対するマルチプロセッサシステムによる故障並列法を用いたテスト生成の並列処理では，プロセッサ間の通信方式として，生成したテストベクトルを通信するベクトルブロードキャスト方式と，検出した故障を通信する故障ブロードキャスト方式が提案されている．これら二つの通信方式を用いたテスト生成の並列処理の解析については既に報告している．本稿では，二つの通信方式をワークステーションネットワーク上に実現し，実験を行った結果と先の解析結果との比較及び性能評価について報告する．

Abstract　Parallel processing on a multiple processor system is available to test generation for large logic circuits. The performance of parallel processing for test generation depends on the method of communication among processors. We have already reported the performance analysis for two types of parallel processing which differ in communication methods, *Vector Broadcasting* and *Fault Broadcasting*. We implemented the above parallel test generation methods on a network of workstations. In this paper, we present the experimental results, compare the experimental results with the analytical results, and evaluate the performance of the implemented systems.

1　はじめに

今日の大規模集積化された論理回路に対するテスト生成処理の高速化は重要な課題である．マルチプロセッサシステムによるテスト生成の並列処理は，高速化の一手法として注目されており，中でも，故障集合を分割して各プロセッサに割り当てる故障並列法が多く研究されている [1][2][3].

故障並列法では，それぞれのプロセッサで異なる故障に対してテストベクトルを生成する．一般に 1 つのテストベクトルで複数の故障が検出されるため，あるプロセッサのテスト生成の目標となった故障が他のプロセッサで生成されたテストベクトルで検出可能なことがある．しかし，たとえそのような場合でも処理は並列に行われているため，プロセッサはその故障に対するテスト生成を実行してしまう．その結果，生成されるテストベクトル数は増加し，また，並列処理によるスピードアップの効果は下がってしまう．

並列化による無駄なテストベクトル生成を抑えるため，生成したテストベクトルをプロセッサ間で交換するベクトルブロードキャスト方式 [2] と検出した故障をプロセッサ間で交換する故障ブロードキャスト方式 [3] が提案されている．ブロードキャストによる情報交換は，頻度を高く，また回数を多くするほど，生成するテストベクトル数を少なくし，高いスピードアップを得ることができると考えられる．しかし，ブロードキャストを多く行うと通信費用は増加し，システム全体の処理時間の低下を招くので，通信費用を低く抑えてブロードキャストの効果を高めるために二つのパラメータ，**通信終了係数**と**通信周期**を考える必要がある．

テスト生成の並列処理の性能 (生成されるテスト

†Performance Evaluation of Parallel Processing for Test Generation
Takaharu FUJII, Tomoo INOUE, Hideo FUJIWARA
Graduate School of Information Science
Nara Institute of Science and Technology

ベクトル数，全体の処理時間) は，これらの通信方法によって大きく変化する．従って，テスト生成における並列処理で高い性能を得るためには，効率の良い通信を行うことが重要な課題となる．

そこで，本研究ではこの点に着目し，通信方法の異なる二つの並列処理方式について，その性能を解析と実験から評価する．性能についての解析は既に行っている [4]．実験は，ワークステーションネットワーク上で行い，二つの通信方式について，プロセッサ数，通信終了係数，通信周期を変化させて，生成されるテストベクトル数，ユニプロセッサシステムに対するスピードアップ率，通信費用を測定した．この実験結果をもとに解析との比較を行い，並列処理の性能について評価を行う．

以下，2 章で故障並列法を用いたテスト生成の並列処理について示し，解析結果を示す．3 章では，ISCAS '85[5], ISCAS '89[6] ベンチマーク回路を用いて実験を行った結果と解析結果と照らし合わせて性能の評価を行う．ただし，ISCAS '89 は，スキャン設計を仮定し組合せ回路に変換したものを用いた．4 章でまとめを述べる．

2 テスト生成における並列処理

本研究では，組合せ回路を対象としたゲートレベルでの単一縮退故障を扱う．

故障並列法を用いたテスト生成の並列処理においては，図1に示すように与えられた故障集合は，部分故障集合に分割され，複数のプロセッサに割り当てる．各プロセッサは，割り当てられた部分故障集合に対してテスト生成を行う．また，各プロセッサは，ネットワークで接続されており，互いに通信して情報交換を行う．

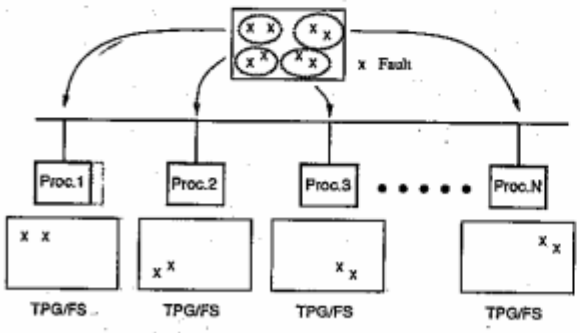

図 1: マルチプロセッサシステム

各プロセッサは，互いに通信して情報交換を行うことによって，無駄なテスト生成を抑える．このプロセッサ間の通信には，前章で述べたように，通信する情報の違いからベクトルブロードキャスト (VB) 方式，故障ブロードキャスト (FB) 方式が提案されている．また，いずれの方式に対しても通信終了係数と通信周期が考えられる．通信終了係数は，故障検出率 p_c で表し，通信はテスト生成開始から故障検出率が p_c になるまで行うとする．通信周期は，生成されるテストベクトル数λ で表し，通信はλ 個のテストベクトルを生成する毎に行うとする．この二つのパラメータを考慮して，以下のようなベクトルブロードキャスト (VB) 方式と故障ブロードキャスト (FB) 方式を考える．

ベクトルブロードキャスト (VB) 方式 プロセッサはテストベクトルを生成すると，そのテストベクトルを用いて，割り当てられた部分故障集合を対象に故障シミュレーションを行う．この処理を繰り返し，生成したテストベクトル数が通信周期λ に達すると，それらλ 個のテストベクトルをブロードキャストする．一方，他のプロセッサからテストベクトルを受けとったプロセッサは，そのテストベクトルを用いて故障シミュレーションを行う．故障検出率が p_c達すると通信を終了し，それ以降は通信を行わずに残存故障がなくなるまでテスト生成を続ける．

故障ブロードキャスト (FB) 方式 プロセッサは一個の故障に対するテストベクトルを生成すると，そのテストベクトルを用いて自分に割り当てられた故障だけでなく，全体の残存故障を対象として故障シミュレーションを行なう．λ 個のテストベクトルを生成し，故障シミュレーションを行った後，検出された故障をブロードキャストする．また，故障を受け取ったプロセッサは残存故障の情報を更新する．故障検出率が p_c に達すると通信を終了し，それ以降は通信を行わずに残存故障がなくなるまでテスト生成を続ける．

これらの通信方法を用いた並列処理の性能についての解析は既に行っている [4]．ここでは，生成され

るテストベクトル数，ユニプロセッサに対するスピードアップ率及び通信費用についての解析結果のみを示す．

2.1 生成されるテストベクトル数

ブロードキャストを行わない NB 方式で得られる総テストベクトル数を $T_{size}(NB)$ とすると，VB 方式で得られる総テストベクトル数 $T_{size}(VB)$ と FB 方式で得られる総テストベクトル数 $T_{size}(FB)$ は等しく，

$$\begin{aligned} T_{size}(VB) &= T_{size}(FB) \\ &= T_{size}(NB) - N\left(1-\frac{\lambda}{(\lambda N)^r}\right)\frac{\log(1-p_c)}{\log q} \end{aligned} \quad (1)$$

と表せる．ただし，q は定数である $(0<q<1)$．この式から，テストベクトルまたは，故障シミュレーションで検出した故障をブロードキャストすることによってブロードキャストを行わない NB 方式に比べて生成されるテストベクトル数を削減できることが分かる．また，通信終了係数が大きいほど，あるいは通信周期が小さいほど得られるテストベクトル数を小さくできることがわかる．

2.2 ユニプロセッサシステムに対するスピードアップ率

テストベクトル生成に要する費用は，生成されたテストベクトル数に比例すると考えられ，式 (1) に示すように VB方式，FB方式とも同数のテストベクトルを生成することから生成に要する費用も等しくなる．一個のテストベクトルを生成するのに必要な費用を c_t とすると，VB方式，FB方式でのテストベクトル生成に要する費用 $C_{test}(VB)$，$C_{test}(FB)$ は，

$$\begin{aligned} C_{test}(VB) &= C_{test}(FB) \\ &= C_{test}(NB) - c_t\left(1-\frac{\lambda}{(\lambda N)^r}\right)\frac{\log(1-p_c)}{\log q} \end{aligned} \quad (2)$$

と表すことができる．ブロードキャストを行うことによって生成されるテストベクトル数が減少するので，生成に要する費用も減少することがわかる．

一方，故障シミュレーションに要する費用 $C_{sim}(VB)$，$C_{sim}(FB)$ は，一個の故障をシミュレーションするのに必要な費用を c_s とし，定数 c_{fs} を用いて，

$$\begin{aligned} C_{sim}(VB) &= C_{sim}(NB) \\ &+\frac{c_sM}{N}\left(\frac{p_c}{1-q^{(\lambda N)^r}}\left(\lambda N-(\lambda N)^r\right)q^{(\lambda N)^r}\right) \\ &+c_{fs}\left((\lambda N)^{1-r}-1\right)\frac{\log(1-p_c)}{\log q} \end{aligned} \quad (3)$$

$$\begin{aligned} C_{sim}(FB) &= C_{sim}(NB) \\ &+c_sM\left(\frac{p_c}{(1-q^{(\lambda N)^r})(1-q)}-\frac{1}{N}\frac{p_c}{(1-q)}\right) \\ &-c_{fs}\left(1-\frac{\lambda}{(\lambda N)^r}\right)\frac{\log(1-p_c)}{\log q} \end{aligned} \quad (4)$$

となる．VB 方式の場合，式 (3) において，

$$\frac{c_sM}{N}\left(\frac{p_c}{1-q^{(\lambda N)^r}}\left(\lambda N-(\lambda N)^r\right)q^{(\lambda N)}\right)>0 \quad (5)$$

$$c_{fs}\left((\lambda N)^{1-r}-1\right)\frac{\log(1-p_c)}{\log q}>0 \quad (6)$$

であることから，故障シミュレーションに要する費用 $C_{sim}(BV)$ は，ブロードキャストを行わない場合の故障シミュレーション費用 $C_{sim}(NB)$ と比べて，ブロードキャストによって受けとったテストベクトルに対する費用だけ増加することがわかる．また，FB 方式の場合，式 (4) において，

$$\begin{aligned} &c_sM\left(\frac{p_c}{1-q^{(\lambda N)^r}(1-q)}-\frac{1}{N}\frac{p_c}{(1-q)}\right) \\ &\qquad<c_{fs}\left(1-\frac{\lambda}{(\lambda N)^r}\right)\frac{\log(1-p_c)}{\log q} \end{aligned} \quad (7)$$

となるとき，FB 方式における故障シミュレーションに要する費用 $C_{sim}(FB)$ は，ブロードキャストを行わない NB 方式の場合よりも小さくなるといえる．

VB 方式，FB 方式を比較すると式 (3)，式 (4) より，

$$\begin{aligned} &c_sM\left(\frac{p_c}{1-q^{(\lambda N)^r}}\left(\left(\lambda N-(\lambda N)^rq^{(\lambda N)^r}\right.\right.\right. \\ &\qquad\left.\left.\left.-\frac{N}{(1-q)}\right)-\frac{p_c}{(1-q)}\right)\right) \\ &\qquad>c_{fs}\frac{\lambda}{(\lambda N)^r}(1-N)\frac{\log(1-p_c)}{\log q} \end{aligned} \quad (8)$$

となるとき，FB 方式における故障シミュレーションに要する費用 $C_{sim}(FB)$ は，VB 方式における故

障シミュレーションに要する費用 $C_{sim}(VB)$ よりも小さくなる．

次に通信費用について示す．一回のベクトルブロードキャストに必要な費用を c_{cv}，故障ブロードキャストに必要な費用を c_{cf} とすると，VB 方式，FB 方式で必要な通信費用 $C_{com}(VB)$，$C_{com}(FB)$ は，

$$C_{com}(VB) = c_{cv}N(N-1)\frac{1}{(\lambda N)^r}\frac{log(1-p_c)}{\log q} \quad (9)$$

$$C_{com}(FB) = c_{cf}N(N-1)\frac{1}{(\lambda N)^r}\frac{log(1-p_c)}{\log q} \quad (10)$$

と表すことができる．

以上から，ベクトルブロードキャスト (VB) 方式，故障ブロードキャスト (FB) 方式それぞれの総費用 $C_{total}(VB)$，$C_{total}(FB)$ は，

$$\begin{aligned} C_{total}(VB) &= C_{test}(VB) \\ &\quad +C_{sim}(VB)+C_{com}(VB) \end{aligned} \quad (11)$$

$$\begin{aligned} C_{total}(FB) &= C_{test}(FB) \\ &\quad +C_{sim}(FB)+C_{com}(FB) \end{aligned} \quad (12)$$

によって得られ，また，ユニプロセッサシステム UP に対するスピードアップ率 $S(VB)$，$S(FB)$ は，

$$S(VB) = \frac{C_{total}(UP)}{C_{total}(VB)} \quad (13)$$

$$S(FB) = \frac{C_{total}(UP)}{C_{total}(FB)} \quad (14)$$

によって得られる．

2.3 通信費用

VB 方式における，一回当たりのブロードキャストに必要な通信費用 c_{cv} は，定数 c_{v0}, c_{v1} を用いて，

$$c_{cv} = c_{v0} + c_{v1}\lambda \quad (15)$$

と表すことができる．一方，FB 方式では，一回当たりのブロードキャストに必要な通信費用 c_{cf} は，通信回数 b，定数 c_{f0}, c_{f1} を用いて，

$$c_{cf} = c_{f0} + c_{f1}M(1-q^{\lambda})q^{(\lambda N)^r(b-1)} \quad (16)$$

となる．ブロードキャスト全体の費用 $C_{com}(VB)$，$C_{com}(FB)$ はそれぞれ，

$$\begin{aligned} C_{com}(VB) &= \\ &(c_{v0}+c_{v1}\lambda)N(N-1)\frac{1}{(\lambda N)^r}\frac{\log(1-p_c)}{\log q} \end{aligned} \quad (17)$$

$$\begin{aligned} C_{com}(FB) &= \\ &\left(c_{f0}\frac{1}{(\lambda N)^r}\frac{\log(1-p_c)}{\log q}\right. \\ &\left.+c_{f1}\frac{p_cM}{1-q(\lambda N)^r}(1-q^{\lambda})\right)N(N-1) \end{aligned} \quad (18)$$

となる．通信回数 b に対して，c_{cv}は一定 (式 (15))，c_{cf}は単調減少関数 (式 (16)) なので，

$$\begin{aligned} c_{f0} + \frac{c_{f1}M(1-q^{\lambda})}{q^{(\lambda N)^r}} &< c_{v0}+c_{v1}\lambda \\ &< c_{f0}+c_{f1}M(1-q^{\lambda}) \end{aligned} \quad (19)$$

が成立するならば，ブロードキャストするデータをテストベクトルから故障に切り替えることによって，通信費用を小さくすることができる．そして，通信回数 b が，

$$B_s = \frac{1}{(\lambda N)^r \log q}\log\left(\frac{c_{v0}+c_{v1}\lambda-c_f0}{c_{f1}M(1-q^{\lambda})}\right)+1 \quad (20)$$

で表される B_s に達するまではベクトルブロードキャストを行い，それ以降は故障ブロードキャストを行うことで，最小の通信費用を得ることができる．

3 実験結果

実験は，ISCAS '85[5]，ISCAS '89[6] ベンチマーク回路を用いて，DECstation 5000 のワークステーションネットワーク上で行った．ただし，ISCAS '89 は，スキャン設計を仮定し組合せ回路に変換したものを用いた．テスト生成には FAN[7] を用いた．

VB 方式，FB 方式それぞれについて，プロセッサ数，通信終了係数，通信周期の変化に対する，1) 生成されるテストベクトル数，2) ユニプロセッサシステムに対するスピードアップ率，3) 通信費用，について測定を行った．

3.1 生成されるテストベクトル数

c7552 に対して，通信終了係数0.8，通信周期 1 としたときのプロセッサ数と生成されるテストベクトル数の関係を図 2に示す．

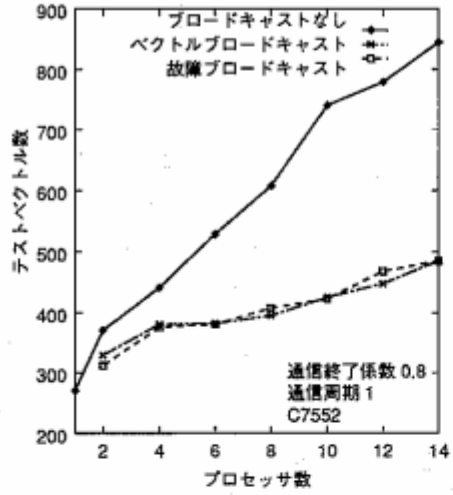

図 2: プロセッサ数とテストベクトル数 (c7552)

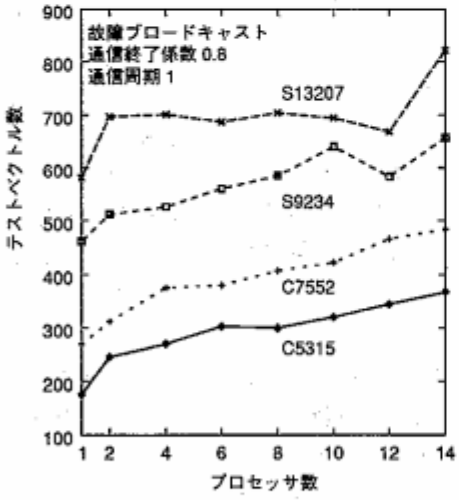

図 3: プロセッサ数とテストベクトル数
(故障ブロードキャスト方式)

ブロードキャストを行わない NB 方式，ベクトルブロードキャスト VB方式，故障ブロードキャスト FB方式いずれの場合も，プロセッサ数が増加すると生成されるテストベクトル数も増加している．並列度が大きくなることによって，無駄なテスト生成が増加するためといえる．しかし，VB 方式，FB 方式では，ブロードキャストを行って各プロセッサが情報を交換することで，ブロードキャストを行わない方式に比べて，プロセッサ数の増加に対する生成されるテストベクトル数の増加を抑えることができている．また，VB 方式，FB 方式ともほぼ同数のテストベクトルが得られている．これらの結果は，解析結果と一致している．図3にいくつかの回路についての故障ブロードキャスト方式を用いたときのプロセッサ数に対する生成されるテストベクトル数を示す．どの回路に対しても通信を行なうことで，プロセッサ数の増加に対するテストベクトル数の増加は少ない．

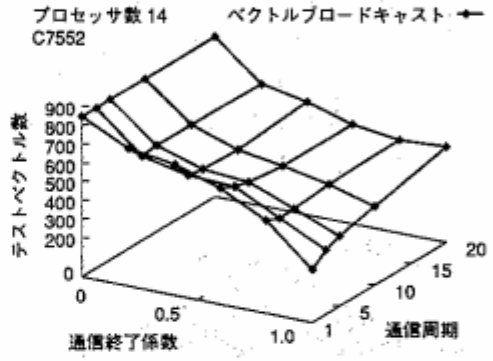

図 4: 通信終了係数・通信周期とテストベクトル数
(c7552 ベクトルブロードキャスト (VB) 方式)

図 4は，プロセッサ数を 14 とし，ベクトルブロードキャスト方式を用いた場合の通信終了係数，通信周期，生成されるテストベクトル数の関係を示している．通信終了係数を大きくすると，得られるテストベクトル数は小さくなる．通信周期を大きくした場合は，生成されるテストベクトル数も増加している．また，通信終了係数 0.2 ～0.8，通信周期が 3 ～10 のとき，少ないテストベクトルが得られている．通信周期が 1 のときに最小のテストベクトルが得られるとする解析結果とは一致しない．これは，通信周期を小さくするとネットワークが混雑し通信費用が大きくなることと，c7552 のような比較的小さな回路ではテスト生成にかかる費用が小さいため，通信周期に達してブロードキャストを行った後，次のブロードキャストを実行するまでに他のプロセッサ

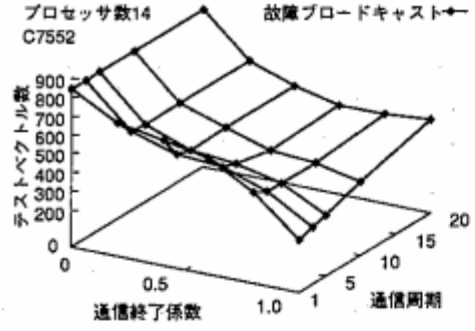

図 5: 通信終了係数・通信周期とテストベクトル数 (c7552 故障ブロードキャスト (FB) 方式)

からのテストベクトルあるいは検出された故障を受けとることができないためと考えられる．FB 方式の場合も同様の結果が得られている (図 5)．c7552 よりも大きな回路 s13207 では，テストベクトルを生成するのに要する費用は大きいので，その傾向は小さくなっている (図 6)．

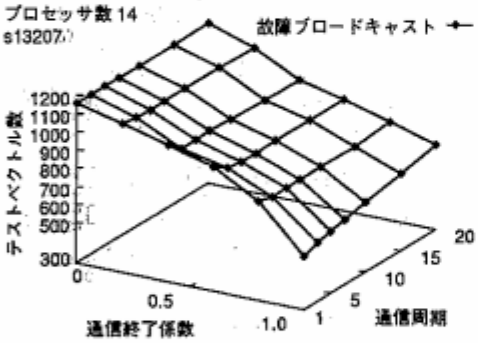

図 6: 通信終了係数・通信周期とテストベクトル数 (s13207 故障ブロードキャスト (FB) 方式)

3.2 ユニプロセッサシステムに対するスピードアップ率

通信終了係数 0.8 ，通信周期 1 とした場合のプロセッサ数とスピードアップ率の関係を図 7に示す．FB 方式，VB 方式ともプロセッサ数が 1 ～8 では，大きなスピードアップ率を得ることができている．しかし，それ以上のプロセッサ数では高いスピードアップ率は得られない．これは，プロセッサ数が大きくなるとプロセッサ間の通信量が多くなって通信費用が増加すること，並列度が増すことによって生成されるテストベクトル数が増加しテスト生成に要する費用も増加するためと考えられる．これらは，解析結果と一致している．次に，VB 方式と FB 方式を比較すると，VB 方式よりも FB 方式の方が高いスピードアップ率が得られている．2 章の解析結果で示したように VB 方式の場合，他のプロセッサから受けとったテストベクトルについても故障シミュレーションを行うので，その故障シミュレーションに要する費用は増加する．一方，FB 方式の場合，受けとった故障を故障リストから取り除く処理に要する費用は小さい．そのため，VB 方式よりも FB 方式の方が高いスピードアップ率が得られたと考えられる．

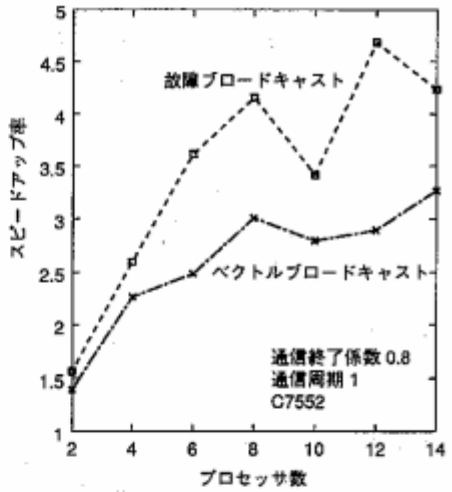

図 7: プロセッサ数とスピードアップ率

図 8に，FB 方式を用い，通信周期を 1 としたときの通信終了係数とスピードアップ率の関係をプロセッサ数毎に示す．どのプロセッサ数の場合も通信終了係数が 0.6 ～0.8 のときに高いスピードアップ率が得られている．特に，プロセッサ数が 8 のとき，ブロードキャストを行うことによって，ブロードキャストを行わない通信終了係数 0 のときよりも高いスピードアップ率を得ることができている．ブロードキャストを多く行うことによる通信費用の増加よりも，通信によって無駄なテストベクトル生成が減ることによるテスト生成に要する費用の減少の方が大きいためと考えられる．通信終了係数が 0.8 ～1.0 で

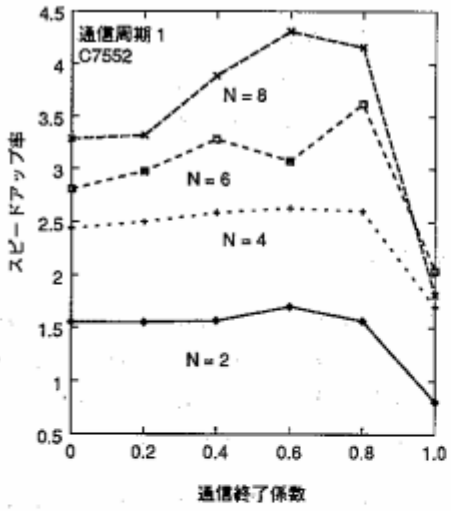

図 8: 通信終了係数とスピードアップ率

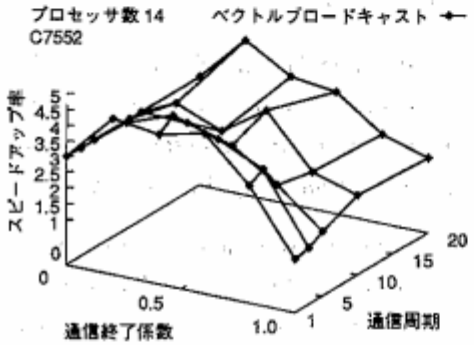

図 9: 通信終了係数・通信周期とスピードアップ率
(ベクトルブロードキャスト (VB) 方式)

は，通信費用の増加が大きいために，どのプロセッサ数でもスピードアップ率は低下している．

図 9と図 10は，VB 方式，FB 方式を用いた場合の通信終了係数，通信周期とスピードアップ率の関係を示している．前節で，通信終了係数を 1.0 ，通信周期を 1 とした場合に，得られるテストベクトル数は最小となることを示したが (図 4，図 5，図 6)，スピードアップ率は，通信費用が増大するために低くなる．また，通信周期が 10 より大きい場合，通信終了係数が 0.2 のときに高いスピードアップ率が得られている．通信周期を 3 ～10 としたときには，通信終了係数が 0.6 ～0.8 のとき，生成されるテストベクトル数も少なく，高いスピードアップ率を得

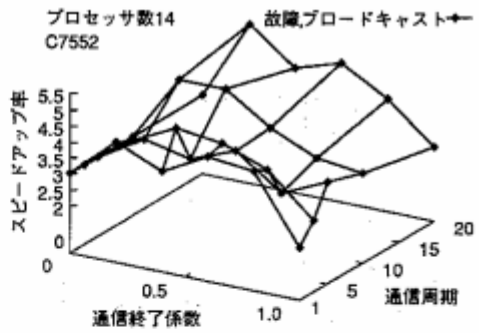

図 10: 通信終了係数・通信周期とスピードアップ率
(故障ブロードキャスト (FB) 方式)

られることがわかる．

通信終了係数を大きくすると生成されるテストベクトル数は小さくなるが，スピードアップ率は小さくなる．通信周期を 1 に設定すると得られるテストベクトル数は大きくなりスピードアップ率も小さくなる．また，通信周期を大きくするとスピードアップ率は高くなるが生成されるテストベクトル数は増加する．このため，ベクトル数の増加を低く抑えて最速な処理時間を得ることのできる通信終了係数と通信周期が存在するといえる．c7552 の回路においてプロセッサ数が 14 の場合，生成されるテストベクトル数の増加をユニプロセッサシステムのときの 2 倍に抑えて，最も高いスピードアップ率を得るためには，通信終了係数を 0.6, 通信周期を 3 としたときである．

3.3 通信費用

通信累積回数に対する 1 回あたりのブロードキャストにかかる通信費用を調べたグラフを図 11 に示す．VB 方式の場合通信費用はほぼ一定である．一方，FB 方式の場合，テスト生成の処理が進むにつれて通信費用は減少している．この実験結果は，解析結果と一致している．また，グラフからもわかるように，VB 方式と FB 方式で，通信費用の交差する点が存在するので，解析で示したように，その点で通信方式を変えることで最小の通信費用を得ることができるいえる．c7552 において通信終了係数 0.8 ，通信周期 1 とした場合には，通信回数が 36 に達したとき通信方式を VB 方式から FB 方式に切替え

ることで最小の通信コストが得られると考えられる．

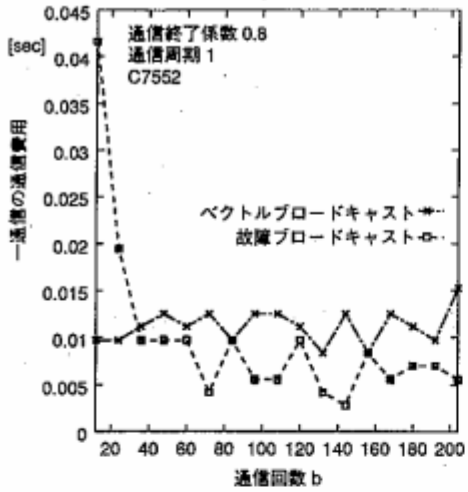

図 11: 一通信に要する通信費用

4 あとがき

故障並列法を用いたテスト生成の並列処理においてその性能を解析と実験から評価した．

実験は，ワークステーションネットワーク上で行い，それぞれのブロードキャスト方式について，プロセッサ数，通信終了係数，通信周期を変化させて，生成されるテストベクトル数，ユニプロセッサシステムに対するスピードアップ率，通信費用を測定した．その結果は，1) 通信終了係数が増加すると生成されるテストベクトル数は減少する，2) 生成されるテストベクトル数が最小となる通信周期が存在する，3) 故障ブロードキャスト (FB) 方式は，ベクトルブロードキャスト (VB) 方式よりも高いスピードアップ率を得ることができる，4) 通信費用は，故障ブロードキャスト (FB) 方式ではテスト生成が進むにつれて減少し，ベクトルブロードキャスト (VB) 方式では一定となる，ことを示した．

これらの実験結果は，解析結果とほぼ一致した．

故障並列法を用いたテスト生成の並列処理において，故障ブロードキャスト (FB) 方式を用い，通信終了係数を 0.6 ～0.8，通信周期を 3 ～10 としたとき，プロセッサ数の増加に対する生成されるテストベクトル数の増加をできるだけ抑えると同時に高いスピードアップ率を得られることがわかった．

今後，より大規模な回路についても実験を行う．また，通信費用を最小にするために通信方式をベクトルブロードキャスト (VB) 方式から故障ブロードキャスト (FB) 方式に切替える手法についても実験を行う．

参考文献

[1] H. Fujiwara, and T. Inoue: "Optimal granularity of test generation in a distributed system," in *IEEE Trans. Computer Aided Design*, vol. 9, no. 8, pp. 885–892 (Aug. 1990).

[2] S. Patil, and P. Banerjee: "Performance tradeoffs in a parallel test generation/fault simulation environment," in *IEEE Trans. Computer Aided Design*, vol. 10, no. 12, pp. 1542–1558 (Dec. 1991).

[3] R. H. Kelenke, L. Kaufman, J. H. Aylor, R. Waxman, and P. Narayan: "Workstation based parallel test generation," in *Proc. 1993 Int. Test Conf.* , pp. 77–84 (1993).

[4] T. Inoue, T. Fujii, and H. Fujiwara: "On the Performance Analysis of Parallel Processing for Test Generation," in *Proc. 3rd Asian Test Symp.* , pp. 69–74 (Nov. 1994).

[5] F. Brglez, and H. Fujiwara: "A neutral netlist of ten combinational benchmark circuits and a target translator in FORTRAN," (Special Session on ATPG and Fault Simulation), in it Proc. IEEE Int. Symp. Circuits and Systerms (Jun. 1985).

[6] F. Brglez, D. Bryan, and K. Kozminski: "Combinational profiles of sequential benchmark circuits," in *Proc. Int. Symp. on Circuits and Systems*, pp. 1929–1934 (Jun. 1989)

[7] H. Fujiwara and T. Shimono: "On the acceleration of test generation algorithms," in *IEEE Trans. Conput.*, vol C-32, no. 12, pp. 1137–1144 (Dec. 1983)