

## 機能縮小による耐故障性プロセッサ

中村 洋介<sup>†</sup> 平木 敬<sup>†</sup>

<sup>†</sup> 東京大学大学院情報理工学系研究科コンピュータ科学専攻  
〒 113-8654 東京都文京区本郷 7-3-1  
E-mail: †{nakmura, hiraki}@is.s.u-tokyo.ac.jp

**あらまし** 本稿では固定故障が発生する状況において非常に長期間使用することが可能である CPU について議論する。故障が発生しても使用し続ける事を前提とした場合にこれに対する従来の解決手法として回路の多重化と回路の代用が挙げられるがそれぞれ問題を抱えている。我々は代用を用いた回路の再構成手法を提案した。提案手法は予備評価において有用であるという結論が得られた。一方、配線遅延による速度への影響、故障部位の再構成手法への影響を無視することはできない。また部分のみに対する再構成手法ではなく、CPU 全体に対する再構成システムを評価を必要がある。本稿では部品に対する位置情報を組み込むことによりこれらの問題を解決し、より詳細な再構成手法を評価する。位置情報を組み込んだプログラム実行時間と故障部位による影響をシミュレーションにより評価した。

**キーワード** CPU 固定故障 FPGA 再構成

## Fault-tolerant Processor by Degradation

Yousuke NAKAMURA<sup>†</sup> and Kei HIRAKI<sup>†</sup>

<sup>†</sup> The University of Tokyo, Graduate School of Information Science and Technology  
Hongo 7-3-1, Bunkyo-ku, Tokyo, 113-8654 Japan  
E-mail: †{nakmura, hiraki}@is.s.u-tokyo.ac.jp

**Abstract** In this paper, we will discuss about repairing of permanent fault. When it is impossible to replace for correct one, it become to be difficult. To resolve this problem, duplication method and substitution method are proposed as previous works. We proposed re-constitution method. This method intend to degrade a function gradually in a CPU when faults occur. We make preliminary simulation about re-constitution method. It is difficult to declare that former simulation indicated that re-constitution method is useful, while it is better to more precise simulation to conclude. In this paper, we will make simulation and more precise result.

**Key words** CPU, Permanent Fault, FPGA Re-constitution

### 1. ま え が き

コンピュータシステムを進化させ続ける事は我々の大きな命題の一つである。進化させるべきその性能の指向は様々である。その一つである、より安定かつより速いコンピュータアーキテクチャを開発することは重要である。我々は非常に長期間動作が可能である CPU について研究している。本稿では特に CPU に固定故障が発生しても非常に長い期間使用が可能であり、また故障が発生した状態でも使用可能である速度性能を持つアーキテクチャについて議論する。

この命題に対する従来の解決手法として回路の多重化 [11] と回路の代用 [2] が挙げられる。回路の多重化は回路を多重化しておき、故障時には正常回路と交換する手法である。回路の代

用は壊れた機能を別の正常な機能を駆使して故障した機能を代用させる手法である。ただし前者は多数故障時に局所的な予備多重化回路欠乏の発生、後者は多数故障下におけるプログラム実行速度性能を考慮した場合に困難であるという問題を抱える。

我々は代用を用いた回路の再構成手法を提案した [15]。この手法は故障の発生とともに徐々に CPU の各機能を縮小することにより急激ではなく徐々に速度性能を低下させる手法である。また多数の故障時には再構成を行い、速度性能低下を低減する手法である。

この手法に対して予備評価を行ったものの部品間配線遅延、CPU における故障発生部位の速度性能への影響を考慮する必要がある。また予備評価における再構成手法は ALU の部位に

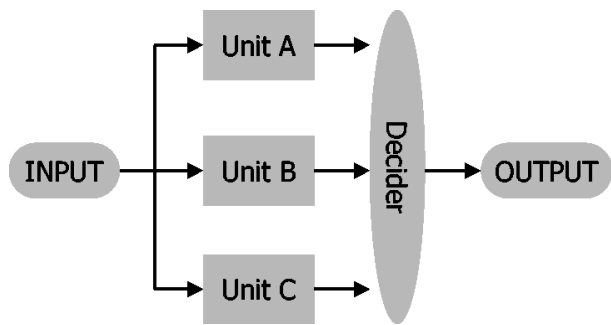


図1 回路の重複:Triple Modular Redundancy

対してのみに評価がなされおり、CPU 全体に対しては評価がなされていない。これらの点をふまえたより評価な評価が必要である。

本稿ではそれぞれの部品に位置という概念を導入する。各部品の配置や部品間配線を行う。そのことにより部品間配線遅延や故障部位による影響を組み込んだ評価を行う。また CPU 全体を再構成システムに組み込み、予備評価よりも正確な評価を行う。

本稿は以下の構成となっている。第2章では従来手法と予備評価の概要について説明する。第3章では予備評価に対する考察とそれに対する問題について議論する。第4章ではその新たな問題点に対する評価方法について説明する。第5章では結果について、第6章では関連研究について述べ、第7章では本稿のまとめを述べる。

## 2. 準備

### 2.1 故障の種類と従来手法

CPU の故障には様々なものがあるが本稿ではそれらを一時故障と固定故障に分ける。一時故障とは一時的に故障する現象である。具体的には電磁場の乱れ等により内部のステートが一時的な異常を示す事などである。特にシステムの再起動により回復することが出来る故障の事である。固定故障とは内部回路が完全に壊れてしまい、論理回路が別物となったものである。具体的には長期使用により回路の消耗、断線などである。特にシステムの再起動による修復はできない故障の事である。

故障が発生した場合にはそれに対応した修復方法備えておかなければ正しい計算が出来ない。この修復方法を全くペナルティ無しで搭載出来る事が最も望ましいのであるがそれは困難である。そのために修復方法を搭載する代りに性能を犠牲にする必要がある。この事より回路面積を性能とした場合のペナルティを背負う回路の重複と速度性能をペナルティとして背負う回路の代用が従来手法が提案され、我々はそれに対して回路の再構成手法を提案した。これらの三種の手法について簡単に説明する。

回路の重複手法 [11] は部品の多重化を行う手法である。具体的には Triple Modular Redundancy(TMR: 図1) のように部品を三重化し、その投票システムによって出力を決定する手法である。概念的に単純なシステムであるという長所を持つ

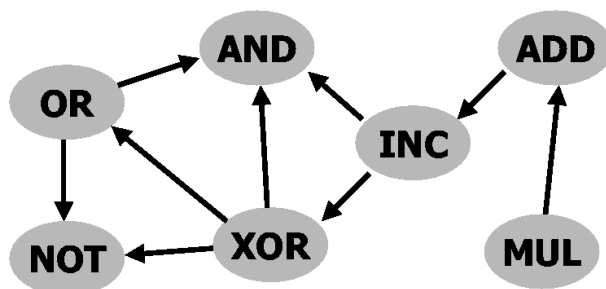


図2 ファンクショナルユニット間の機能代用の依存関係

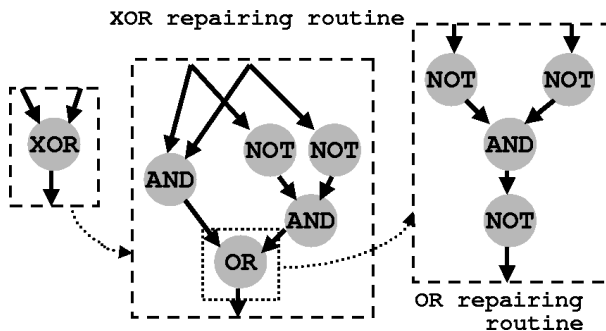


図3 再帰的呼び出しによる故障の修復

方、回路面積の増大や一つの部分で重複化が切れた場合にはいくらか別の部分に予備があっても固定故障を回避することが不可能であるという短所を持つ。

回路の代用手法 [2] は壊れた機能を正常に機能している別機能を駆使してその機能を代用させる手法である。どのような正常な機能を使っても壊れた機能を実現できる事は無く、壊れた機能を修復できる組み合わせは限られている。図2ではファンクショナルユニットにおける故障とその修復関係の依存の例である。具体的には XOR は AND、OR、NOT を使用して修復が可能である例である。

この手法のもう一つの特徴として再帰的呼び出しによる複数の故障の修復が挙げられる。例として図3では XOR が故障していた場合には AND、OR、NOT による修復が考えられるがさらに OR まで故障していた場合には OR を AND、NOT により修復が可能である。

この手法は回路サイズが小さいながら修復手段をとることが可能であるという一方プログラム実行速度を考慮した改良手法をとったものでも多数の故障時では実行速度が低下するため、多数故障時における使用は困難である。

### 2.2 予備評価

回路の代用を用いた再構成手法は故障が発生した場合に CPU の機能縮小と回路の再構成を利用し、故障発生とともに徐々に性能を低下させる手法である。具体的には Functional Block(FB: 後述) 交換、機能代用、再構成を使用して故障修復を行う手法である。

まず再構成手法の最小論理回路単位である Functional Block(FB) について説明する。FB は基本的な論理回路を実現することができる、内部論理を自由に書き換えられる回

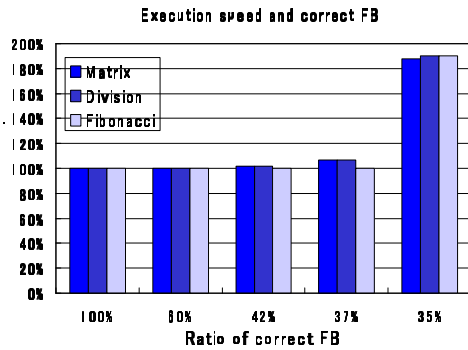


図4 予備評価における実行速度と故障割合の関係

路部品である。また、それら FB 同士は互いにルーティングスイッチで結合されており、その結合関係は初期化に依存するものとする。この FB のそれぞれを基本的な論理回路として使用し、またそれらを任意に結合することにより CPU として機能させることが可能である。具体的には FPGA の CLB などが考えられる。

次に再構成手法の全体的な修復について説明する。まず CPU は FB により構成されている。CPU に故障が発生した場合、別の表現を行えば FB に故障が発生した場合には正常に機能している未使用である FB と交換して修復する。交換が可能である FB が不足した場合には CPU においてあまり重要ではない部品の機能縮小を行い、余剰 FB を作り出す。そこで使用可能になった FB を故障 FB と交換する。残りの FB は予備 FB として次の故障に備えておく。

故障の発生とともに CPU の機能が縮小されていくのだが、縮小された部品は必ずしも単独で修復以前の機能を果たせるわけではない。そのために縮小された機能は場合によっては回路の代用の手法を用いて実行される。

再構成手法において予備評価を行った。再構成手法を ALU 部位に施した CPU 上でフィボナッチ数列、行列演算、除算プログラムを実行した結果が図 4 である。図 4 はプログラム実行時間と故障割合の関係のグラフである。縦軸は全ての FB が正常に機能している場合に対して正規化したプログラム実行時間の比である。横軸は全ての FB に対して正常に機能している FB の割合を示している。例えば 53% の FB が正常に機能している場合にはプログラム実行時間は 1.07 倍であることを示している。故障割合が小さい場合には速度性能低下は小さく、特に故障が増大した場合には急激に速度性能低下は急激に大きくなるのが得られた。この事より故障が多数存在する状況の下では再構成手法は有力な手法であると考えられる。

### 3. 再構成手法と位置情報

再構成手法に対して予備評価を行ったがさらにその評価において詳細に検討した上でその有用性を評価する必要がある。ここでは特に予備評価における仮定、問題点を考慮した上でより

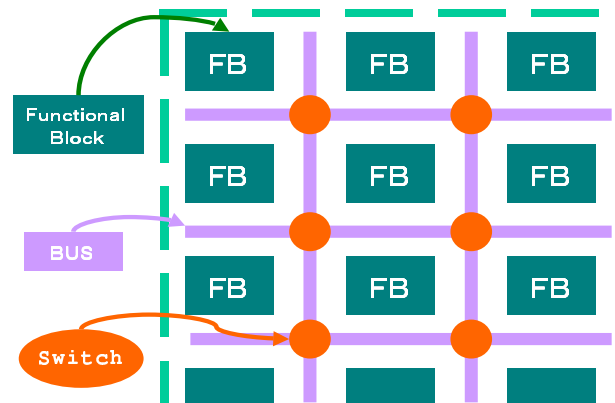


図5 FB マトリクス

正確に有用性を考慮する事について議論する。

#### 3.1 予備評価の問題点

まず配線遅延問題について述べる。プログラム実行速度は CPU のクロックに大きく影響をうける。クロックは回路サイズ、通過すべき配線の長さに影響をうける。再構成手法で FB の交換を行う場合、交換する正常な FB が故障した FB から近い距離に存在するのであればよいのであるが遠く離れた場所に存在する場合もある。この場合には配線遅延が大きくなるためクロックを大きく落とすことになる。また、交換される FB により速度性能にばらつきがあることも考えられる。再構成手法ではより正確な速度性能評価を行うためにこの位置における総合配線ディレイを考慮しなければならない。予備評価において全ての部品間距離は全て一定であるという仮定であり、この点においてさらに改良を行う必要がある。

次に故障部位問題について述べる。故障部位が再構成手法の評価に大きく影響を与えることは前述の通りであるが、同様に故障部位によりその評価に影響を与えることも考慮しなければならない。具体的には故障が周囲に集中する場合と故障が中央より発生する場合には配線ディレイや配置による配線ディレイのばらつきに大きく影響をあたえられられるからである。また故障部位により使用不可能である FB の存在も考えられるため、故障部位による影響も考える必要がある。予備評価においては FB の位置という概念が存在しないためにこの故障位置による評価を行っていない。

最後に再構成手法適用部位の問題について述べる。再構成手法を使用する場合、通常 CPU において機能縮小を適用しやすい機能と適用が難しい機能が存在する。具体的にはキャッシュメモリはそのサイズ縮小により機能縮小を行いやすいが、コントロールシグナル計算用回路等は機能縮小により余剰 FB を作り出すことは難しい。予備評価においては機能縮小が可能である ALU のみに適用して評価をおこなった。より詳細な評価を行うためには機能縮小の困難である部位を含んだ上で評価を行う必要がある。

#### 3.2 FB マトリクス

予備評価の問題点について述べたがこれらを改良するためには評価の上で部品に位置の概念を導入する必要がある。このた

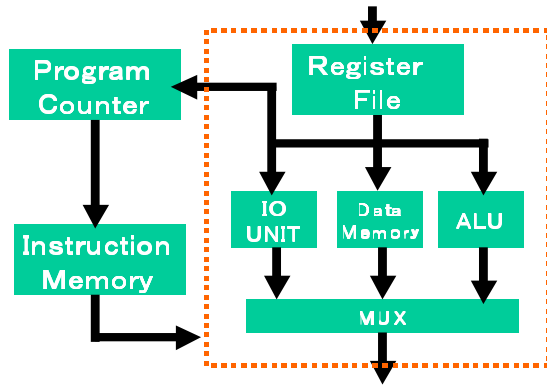


図6 CPUの概要

め我々はFBマトリクスを導入する。これはFBをマトリクス状に配置したCPU位置情報モデルである(図5)。このFBマトリクス上ではそれぞれのFB間にバスが配置されており、またそのバスの交差点ではルーティングスイッチが配置された構造になっている。FBから隣接するルーティングスイッチまで接続されている構造である。このように位置モデルを導入し、この上にFBを配置、配線することによりFB間の位置、配線遅延情報を組み込むものとする。またこのためにCPU全体をFBで構成し、その再構成可能部位も不可能部位も組み込むことができ、その評価を行うことができる。

#### 4. 評価手法

我々はCPUの部品各部に位置情報を持たせるという形により詳細な再構成手法の評価を行う。評価にはCPUのプログラム実行を評価するためのハードウェアシミュレーションプログラム、そのCPU構成部品に位置情報を持たせるFBマトリクスシミュレーションプログラム、コンパイラ、実行プログラムからなるアプリケーションプログラムを使用する。

##### 4.1 CPUについて

CPUは基本的な32ビットRISCアーキテクチャでパイプライン、スーパースカラー、分岐予測、投機的実行は搭載していない。CPUはプログラムカウンタ(PC)、インストラクションメモリ(IM)、レジスタファイル(RF)、算術ユニット(ALU)、データメモリ(DM)、I/Oユニット(IOU)からなる。メモリはそれぞれ1サイクルの読み書き込みが可能である。レジスタファイルは32ビット32本、ツーポートである。

##### 4.2 FBマトリクスについて

FBマトリクスは位置情報を組み込むための仮想CPU基盤モデルである。FBマトリクスではFB、フリップフロップ(FF)、ルーティングスイッチ、バスが定義されてある。

評価においてFBは3入力1出力のルックアップテーブル(LUT)であるとする。このLUTは初期化時にAND、NOTなどの基本論理回路の形にフォーマットされ使用される場合や、実行時にデータを書き換えるレジスタの形で使用されたりする。内部でタイミングをとるためにFFを準備した。これは

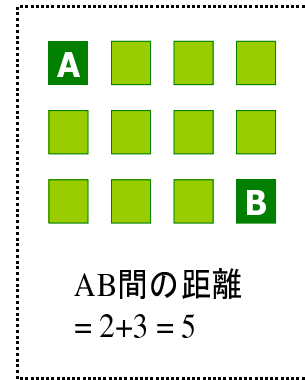


図7 距離の定義

基本的にルーティングスイッチと同じ場所に存在するものとする。バスはそれぞれのFBの行、列の間に網目状に定義されている。そのバスの交差する場所にルーティングスイッチが存在する。バス、ルーティングスイッチの使用状況による占有を考慮するという観点から望ましくはないのであるが、よりモデルをシンプルにするために今回はバス幅等使用する場合の制限は設けないことにする。

評価において距離評価、バス遅延評価を行う必要がある。評価において二つの部品間距離はその座標の差の絶対値(図7)であると定義する。またバス遅延は通過部品による遅延に影響をうけずにそのバスの長さ、特に部品間距離に依存するものとする。

今回の評価においてFBマトリクスは60×60個の計3600個のFBがマトリクス状に定義されているものを使用し、FFは18×18の324個が均等に分散して配置されている。バス、スイッチは前述の通り、特に制限を設けてはいない。

##### 4.3 コンパイラと実行プログラムについて

CPUシミュレーションプログラム上で実行するためのコンパイル手順は次の通りである。まずサンプルプログラムをgcc-2.95.3でSPARCアセンブリ言語にした後、それをCPUシミュレーションプログラム用のバイナリトランスレータで実行プログラムにする。このバイナリトランスレータでアーキテクチャ差違を吸収する。サンプルプログラムとして行列演算、フィボナッチ数列、除算を準備した。

##### 4.4 故障モデル

故障がどの部位に現れるかは配線遅延に影響を与える重要なパラメータである。今回は4種類の発生部位を想定した。

- 片側集中故障

故障がある一辺から発生するモデルである。

隣接する特定部品からの悪影響による故障を想定している。

- 中央集中故障

故障が中央に集中するモデルである。

熱等内部からの悪影響を想定している。

- 周囲集中故障

故障が周囲全般から発生するものを想定している。

悪影響が周囲全般からのものである故障を想定している。

| ALU タイプ   | Function units              | 正常FB の割合 |
|-----------|-----------------------------|----------|
| 32-Shift  | 32-shift 32-add,sub,logical | 100%     |
| 16-Shift  | 16-shift,32-add,sub,logical | 69%      |
| 8-Shift   | 8-shift,32-add,sub,logical  | 52%      |
| 16-AddSub | 4-shift,16-add,sub,logical  | 49%      |

表1 ハードウェアタイプとそれが持つファンクショナルユニット

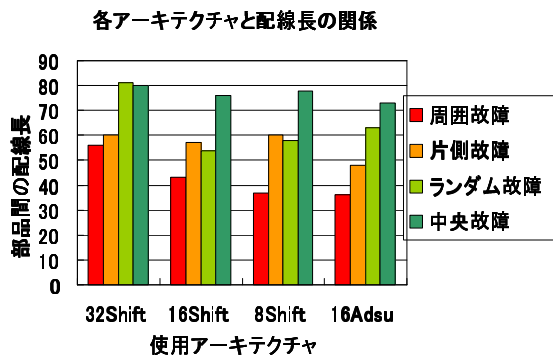


図8 結果: 各故障モデルにおける配線長と機能縮小

- ランダム故障

故障がランダムに発生するモデルを考慮している。

原因は特に特定できないが突発的な故障を想定している。

#### 4.5 再構成部位

再構成手法において機能縮小を行うことの容易である機能と容易ではない機能がある。性能を評価する場合の機能を機能縮小するかについて議論すべき事項であるがこの評価においては ALU の一部の機能が機能縮小をするものとする。四つの機能保持状態 (32-Shift、16-Shift、8-Shift、16-AddSub) を準備した。32 ビットから 8 ビットへのシフトの縮小、続いて 32 ビット加減算器の縮小の順に必要な FB 数は少なくなる (表 1)。

### 5. 結果

まず故障モデルが再構成手法にどのような影響を与えるかについて実験を行う。図 8 はそれぞれの機能所持状態と最大配線長を各故障モデルごとに示したものである。縦軸は部品間距離を示しており、横軸はどの機能がどのような機能縮小を行われているかを示している。故障モデルは左より順に周囲故障モデル、片側故障モデル、ランダム故障モデル、中央故障モデルである。周囲故障モデルでは故障とともに配線遅延が小さくなった。中央故障モデル、片側故障モデルでは変わらず、ランダム故障では特に配線長の変化に傾向が見られなかった。故障モデルにより故障発生による機能縮小と最大配線長において変化の傾向が異なるということが言える。

図 9 は機能縮小によるプログラム実行速度の変化を示している。縦軸は中央故障モデルにおけるプログラム実行速度性能比、横軸は機能縮小される場合のその時所持している機能を

中央故障モデルにおける各アーキテクチャと実行速度比

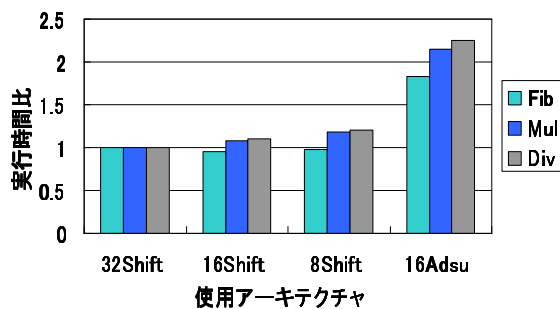


図9 結果: 中央故障モデルにおける速度性能

示している。なお、縦軸は機能縮小されていない Shift-32 の数値で正規化されている。グラフの左よりフィボナッチ数列、行列演算、除算を示している。グラフより初期の機能縮小による速度性能低下は小さく、加減算器の変化のようにプログラムにおいて重要であるユニットの変化の影響が大きく速度性能へと影響していることがわかる。

速度性能評価において再構成手法を全体することにより故障割合と速度性能の数値は異なるものの、故障割合が大きい場合には急激に速度性能が低下するという傾向において予備評価と同様な結果がえられた。また一方位置情報の組み込みによって故障モデルと配線の最長パスの長さの関係を得ることが出来たが、異なる故障モデルにおいてはバス長の変化の傾向はそれぞれ異なる。このことより機能変化に伴う速度性能の変化傾向を見る場合には位置評価による影響は小さい。一方故障モデルを意識した場合には位置情報を組み込む必要があることがわかる。

### 6. 関連研究

[11] では CPU の故障回避のために回路の多重化の手法を使用している。ただし、回路面積が大きくなることが指摘されている。[4] では LEON-1 を提案している。これは長期使用の観点よりピーク性能よりもむしろコンスタントなプログラム実行速度を目指している。[16] ではソフトウェアレベルでの多重化を行い、耐故障性を得ることを提案している。マルチスレッドではない TMR、マルチスレッド TMR、マルチプロセッサ TMR という 3 種のソフトウェアレベルの多重化を行い、再計算や別プロセッサ実行により耐故障性を得ている。[2] では回路の代用手法を提案している。この中では修復手法を組み込むための回路サイズが小さくなることを目指している。回路の代用手法をさらに改良し、動的に代用パスを切り替えによる速度性能の改善を行っているが多数の故障においては快適な速度性能を得るのは難しい。[7], [8] では再構成回路の基本論理単位回路の使用法についてである。単位論理回路を単体の形で使用せず、ある程度ブロック化した形において利用する。このことによりその回路サイズの縮小やコンフィギュレーション

時間短縮、チェック回路を挿入を行っている。実際に再構成手法を使用した CPU を構築する場合にはこれらの技術は有用である。[6], [9], [13] では FPGA デバイスの概要が述べてあり、FPGA の配線資源問題が重要であることが述べられており、本研究のさらに課題を与えるものである。

## 7. ま と め

本研究では CPU の固定故障を修復するため、従来の回路の多重化、回路の代用手法に対し回路の代用を用いた再構成手法を提案した。その予備評価に対し位置情報を加え、CPU 全体における再構成を適用し、より詳細な再構成手法の評価を行った。予備評価と同様多数の故障を想定した使用においては再構成手法は有用であることが得られた。また故障部位による配線遅延を考慮する必要がある場合には位置情報による評価が必要であるという結論が得られた。

## 8. 謝 辞

本研究は 21 世紀 COE 「情報科学技術戦略コア」大域ディベンダブル情報基盤プロジェクトの一環として行われている。

### 文 献

- [1] Actel Corporation. *Accelerator Family FPGAs v2.0(Datasheets, General Purpose FPGAs)*, 2003.
- [2] S. C. Alfredo Benso and P. Prinetto. A self-repairing execution unit for microprogrammed processors. *IEEE/micro*, pages 16–22, Sep-Oct 2001.
- [3] Altera Corporation. *Chapter 2. Stratix Architecture(Stratix Device Handbook, Volume 1, ver 2.2, Section I. Stratix Device Family Data Sheet)*, 2003.
- [4] J. Gaisler. Leon-1 processor - first evaluation results. *European Space Components Conference 2000*, March 2000.
- [5] F. Hanchek and S. Dutt. Methodologies for tolerating cell and interconnect faults in fpgas. *IEEE Trans computer*, pages 15–33, January 2000.
- [6] S. Hauck. The future of reconfigurable systems. *Proceedings of the 5<sup>th</sup> Canadian Conference on Field Programmable Devices*, June 1998.
- [7] W. H. M.-S. J. Lach and M. Potkonjak. Efficiently supporting fault-tolerance in fpgas. *Proc. ACM/SIGDA Intl. Symp. on FPGAs*, pages 105–115, Feb 1998.
- [8] W. M.-S. John Lach and M. Potkonjak. Algorithms for efficient runtime fault recovery on diverse fpga architectures. *Proceedings of the 1999 International Symposium on Defect and Fault Tolerance in VLSI Systems*, 1999.
- [9] S. H. K. Compton. Configurable computing: A survey of systems and software. *Northwestern University, Dept. of ECE Technical Report*, 1999.
- [10] P. C. Kanellakis and A. A. Shvartsman. *Fault-tolerant Parallel Computation*. Kluwer Academic Publishers, 1997.
- [11] T. A. G. L. Spainhower. Ibm s/390 parallel enterprise server g5 fault tolerance: a historical perspective. *IBM Journal of Research and Development*, 43, 1999.
- [12] V. Lakamraju and R. Tessier. Tolerating operational faults in cluster-based fpgas. *In ACM International Symposium on Field-Programmable Gate Arrays*, 2000.
- [13] W. Mangione-Smith and B. Hutchings. Configurable computing: The road ahead. *Reconfigurable Architectures: High Performance by Configure (R. Hartenstein and V. Prasanna, eds.), Microsystems Engineering Series, (Chicago), pp. 81–96, IT Press, 1997. Proceedings of the Reconfigurable Architectures Workshop (RAW '97)*, 1997.
- [14] MIPS Technologies, Inc. *MIPS Technologies, Inc., MIPS R4000/R4400 Microprocessor User Manual, Second Edition*, October 1994.
- [15] Y. Nakamura and K. Hiraki. Highly fault-tolerant fpga processor by degrading strategy. In *2002 Pacific Rim International Symposium on Dependable Computing*, Dec 2002.
- [16] N.Saxena and E. McCluskey. Fault-tolerance with multi-threaded computing - a new approach. *Fast Abstracts FTCS-29*, pages 29–30, June 1999.
- [17] The Open SystemC Initiative (OSCI). *System C Version 2.0 User's Guide, 2000*.
- [18] D. L. Weaver and T. Germond. *The SPARC architecture manual: version 9*. Prentice Hall, Englewood Cliffs, NJ, 1994.
- [19] M. Wirthlin and B. Hutchings. A dynamic instruction set computer. *IEEE Workshop on FPGAs on Custom Computing Machines*, April 1995.
- [20] Xilinx, Inc. *XC9500XV 2.5V High-Performance CPLD Family*, June 2002.
- [21] Xilinx, Inc. *CoolRunner-II CPLD Family*, October 2003.
- [22] Xilinx, Inc. *Spartan-3 1.2V FPGA Functional Description*, July 2003.
- [23] Xilinx, Inc. *Virtex-II Pro X Detailed Functional Description (Module 2)*, November 2003.