

動的再構成可能デバイスの耐故障化に関する検討

越智 直紀[†] 中原健太郎^{††} 森江 太士[†] 神山 真一[†] 泉 知論[†]

越智 裕之[†] 中村 行宏[†]

[†] 京都大学 大学院 情報学研究科 通信情報システム専攻

^{††} 京都大学 工学部 電気電子工学科

〒 606-8501 京都市 左京区 吉田本町

E-mail: reconf@easter.kuee.kyoto-u.ac.jp

あらまし 再構成可能デバイスを用いることで、遠隔地からのシステムの更新や、故障発生に際し故障箇所を避けて回路を再構成することによるシステムの回復などが可能になると期待されており、ハードウェアの保守が困難な宇宙開発などの分野から注目されている。しかし、宇宙空間等の過酷な状況ではメモリ上の構成情報の破損が頻発し、それが実用の妨げとなっている。本研究では、そのような構成情報の破損に耐えうる動的再構成システムの開発を目指す。本稿では、まず再構成可能システムに関連する耐故障化の研究開発の動向を示し、次に動的再構成デバイスの耐故障化について基礎的な検討を行う。デバイス・アーキテクチャとして、多重化と多数決に基づくもの、エラー訂正符号に基づくものを挙げる。これらは、動的再構成能力を積極的に活用することで破損した構成情報を自律的に修復する。キーワード 耐故障、動的再構成可能デバイス、論理故障、エラー訂正、多重化

A Discussion on Fault Tolerance of Dynamic Reconfigurable Device

Naoki OCHI[†], Kentaro NAKAHARA^{††}, Futoshi MORIE[†], Shin'ichi KOUYAMA[†],

Tomonori IZUMI[†], Hiroyuki OCHI[†], and Yukihiro NAKAMURA[†]

[†] Department of Communications and Computer Engineering,
Graduate School of Informatics, Kyoto University

^{††} School of Electrical and Electronic Engineering,
Faculty of Engineering, Kyoto University

Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501 Japan

E-mail: reconf@easter.kuee.kyoto-u.ac.jp

Abstract Reconfigurable logic devices are expected to be key devices for systems in severe environment such as spacecrafts, satellites, nuclear power plants and so on, since a system with such devices can be updated remotely by uploading new configuration data and, even for the case of physical defects, may be recovered by reconfiguration excluding faulty parts. However, in order to utilize reconfigurable devices in such a severe environment, we have to cope with logical defects of configuration data. Our goal is to enhance fault tolerance of dynamic reconfigurable systems especially to logical defects in configuration data. In this paper, we review previous works on fault tolerance of reconfigurable systems. Then, we discuss on frameworks of device architectures to enhance fault tolerance. We propose two types of architectures, one based on majority voting, and the other based on error-correcting codes. Both of them correct faults in configuration data autonomously utilizing the ability of dynamic reconfiguration.

Key words Fault tolerance, Dynamic reconfigurable device, Single event upset(SEU), Error correction, Multiplex

1. はじめに

半導体技術が大きく発展し、コンピューターの中核部分であるLSIは急激な進歩を遂げている。そして現在、FPGA (Field Programmable Gate Array) などの再構成可能 (リコンフィギュラブル) デバイスが、回路の試作や少量生産など多くのシーンで利用されている。リコンフィギュラブルデバイスは、演算器や LUT (Look-Up Table) などを多数配置し、それらの構成情報をユーザが変更することで様々な機能を実現できるデバイスである。この再構成可能デバイスの発展型として、システム稼働中にも構成を書き換えることができる動的再構成 (ダイナミックリコンフィギュラブル) デバイスが開発されており、その実用システムの開発も行われている。現在、それを用いたリコンフィギュラブルコンピューティングと呼ばれる計算機科学分野は、研究が盛んに進められている。

現在、動的再構成可能デバイスを宇宙開発、原子力発電所の制御、海底開発などで利用することに、大きな期待が寄せられている。これらが行われる場所は、高温高圧、真空、放射線、荷電粒子など、機材自体が物理的な損傷を受けかねない過酷な環境である。したがって、稼働開始後にシステム更新や修理が必要になった場合、人間が行って作業を行うことは非常に困難である。CPU を用いる場合、全ての機能は固定のハードウェアで実現され、アプリケーションはソフトウェアで実装される。ソフトウェアの更新でアプリケーションのメンテナンスが可能であるが、性能や消費電力に関して問題がある。ASIC は、特定のアプリケーション向けにのみ全ての機能が実装される。従って、性能や消費電力の点では有利だが、アプリケーションの変更やバグを取り除くなどは非常に困難である。それに対し、再構成可能デバイスではメモリ素子にアプリケーション向けの構成情報を記憶するため、必要に応じて機能を書き換えられる。その「書き換えられる」特長を活かし、構成情報を遠隔地から書き換えることで、システム更新やバグフィクス、故障箇所を避けた再構成による擬似的な修理などが期待される。

しかし、そのような過酷な場所では、非常に高い信頼性が求められるにも関わらず、物理故障 (ハードエラー) と論理故障 (ソフトエラー) が多く発生する。物理故障は常に発生しうる障害で、恒久的な障害となる。それに対し、論理故障は、動作中に起こりうる損傷であり、データの論理値が反転してしまう障害である。この故障は、放射線や電子線などが多く飛び交う宇宙空間などで特に発生する故障として知られる [1]。その頻度はおおよそ数十分～数時間に 1 回程度で、メモリ素子の種類によって、発生頻度は大きく異なる [1]。大まかな発生頻度は、DRAM>SRAM>EEPROM≒Flash Memory である。

再構成可能デバイスにおいては、処理対象のデータだけではなく、その論理や接続の構成情報もメモリ素子に記憶され、それらにも論理故障は発生する。論理の構成情報に論理故障が発生すれば、誤った計算結果が出力される。それだけではなく、通信線の接続構成情報にも論理故障が発生する可能性があり、予期しない接続が構成された場合に故障が発生しているにも関わらず検出できなかったり、致命的な接続の誤り (出力同士が

短絡する場合など) が発生するおそれがある。このような、破損した構成情報によって構成される回路は全く予想できず、システムの重大な障害を引き起こす可能性さえある。「書き換えられる」という長所が、そのまま「書き換わってしまう」という短所に変わってしまい、再構成可能デバイス導入における大きな障害となっている。

本研究では、以上に挙げた問題点を克服し、高い信頼性を持つ動的再構成システムの実現に向けたデバイスの構成を検討し、実現性を明らかにすることを目標とする。そのために我々はまず、動作中に構成情報を書き換えられる動的再構成可能アーキテクチャに着目した。このアーキテクチャを用いることで、論理故障によって構成情報が「書き換わってしまう」という短所を、「動作中に書き直す」ことで克服することを考える。次に、耐故障性を実現するためには、システムが要求する信頼性に応じた冗長性を付加する必要がある。その具体的な手法としては、複数回同じ処理を行うことで時間的な冗長性を高める手法、同じ機能を複数個搭載して同じ演算を行う、または故障を検出するための専用機能を追加するといった空間的な冗長性を高める手法がある。それらの手法を実現するための、障害の検出と特定をするための機構、障害を修復あるいは回避するための機構、障害によって誤った動作をした場合に正常な状態に戻す機構が必要になる。これらの機構を検討し、実装することで耐故障性が向上することを確認する。しかし、これらの機構を付加することによって、性能や重量、チップ面積など様々な条件が悪化してしまうことが予想される。そこで、実現される信頼性と種々のパラメータとのトレードオフ関係を明らかにしていく。

本論文では、再構成可能デバイスにおける耐故障性、特に論理的な情報の破損に対する過去の研究についての調査とそのまとめ、それを踏まえた基礎検討を行い、耐故障性と高い信頼性を持った動的再構成可能デバイス実現に向けての今後の研究の指針について述べる。過去の研究の調査は、再構成可能デバイスに関する研究会で最も歴史ある FPL(The International Conference on Field-Programmable Logic and Its Applications) や、軍事目的や宇宙開発などのために耐故障に関する検討を必要とする EH(NASA/DoD Conference on Evolvable Hardware), MAPLD(Military and Aerospace Applications of PLDs) などを中心に調査を行った。それらの中から主要なものを大まかに分類し、第 2 章で紹介する。そして第 3 章で、紹介した研究と我々の研究との狙いの違いを述べ、動的再構成可能デバイスに耐故障性を持たせるための基礎的な検討について述べる。耐故障性を持たせるための手法として代表的な、システムの多重化を行う手法と、誤り訂正符号を持たせる手法の 2 種類について、基本的な検討を行う。そして、第 4 章で、まとめと今後の指針について述べる。

2. 再構成可能デバイスにおける耐故障研究

再構成可能デバイスにおける障害のうち、物理的故障に関しては、故障箇所を避けて構成するなどといった対処法が考えられる。論理的な故障への対処法としては、原因となる宇宙線や電子線を物理的に防ぐ (鉛を被せるなど)、素子サイズ大きい

旧世代のプロセス技術を用いるなどといった対処法が挙げられる。しかし、このような手法には、重量や大きさ、性能に関して問題があると考えられる。これに加え、耐故障化システムを作る手法を再構成可能デバイスに応用したものととして、デバイス自体を複数用いて多重化する手法、デバイス上に多重化された回路を構成する手法が挙げられる。また再構成可能デバイスの特長を活かして、物理故障に対する代替処理を行う手法が提案されている。さらに、構成情報を確認して訂正する手法が提案されており、これに対しても再構成可能デバイスの特長を積極的に活用する手法も提案されつつある。この章では、これら再構成可能デバイスに対しての耐故障研究の中から、主要なものを大まかに分類して紹介する。

2.1 チップを複数個用いて冗長化するもの

もっとも基本的な冗長化の手段としては、同じ機能を構成したチップを複数個搭載することが考えられる。

Wei-Je Huang(Stanford University) らは [1] など、2つのFPGA チップに互いの状態をチェックする機構を構成することで、耐故障性能を向上させる手法を紹介している。

2.2 同じ機能を複数個構成するもの

FPGA 上に同じ機能を複数構成し、多重化を行って耐障害性を向上させることが考えられている。

J. Moreno(Technical University of Catalunya) などによって 1999 年に EH で発表された [2] では、FIPSOC(Field Programmable System on Chip) というデバイスを用いて、系を 2 重化することで耐障害性を向上させる手法が提案されている。2つのコンテキストを同じ構成にし、これらを用いて同じ計算を 2 度行って出力を比較する。出力が異なる場合は故障を検出したと判断し、片方のコンテキストを修正しながらもう一方で計算を続ける。故障が致命的となる比較ユニットなどに関しては、定期的に再構成を行うことで信頼性を高める。

T. Bartzick(University of Siegen) らは 2000 年に FPL で発表された [3] で、4つのセルを 1 まとめにし、そのうち 3つを同じ機能に構成し、残り 1つをテストすることで、耐故障性を向上させる手法を提案している。再構成を管理する RCU によって、故障の検査は動作中に行われ、検出した場合は FPGA 全体を停止し、故障セルとテスト中のセルを置き換える。テスト中のセルが故障した場合は、そのセルを無視する。

Alexander H. Jackson(The University of York) らは、2003 年に EH で発表した [4] で、FPGA 上に 3 重化と多数決回路を構成することで耐故障性を向上させる手法を提案している。この手法では、修復中でも動作を続けることが可能である。

P.K. Lala(University of Arkansas) らは、2004 年に DFT で発表した [5] において、FPGA を止めることなく論理故障を修復する手法を提案している。4つの CLB を 1 組にしたものを Cell と呼び、そのうち 3つを同じ構成にし、1つをスベアとして相互接続する。3つのうちどれかに故障が起きた際はスベアと交換する。この交換はすぐに行うことができる。以上のようにして、動作を止めずに単一の論理故障に対処する。

2.3 物理的破損に対し代替処理するもの

再構成可能デバイスの構成単位の一部に物理的な損傷が無い

かを診断し、損傷が検出された部分を回避する手法を考えるものである。耐故障という観点だけでなく、[6] で述べられているように、デバイスの歩留まり向上という観点からも重要である。

Prasanna Sundararajan(Xilinx Inc.) などによって、FPGA の CLB(Configurable Logic Block) の一部が物理的欠陥を持っている場合に欠陥部分を避ける手法が、2001 年に発表された [6] で述べられている。Xilinx の FPGA ツールキット JBits を用い、構成情報を入力する際、欠陥を持った CLB を状況に応じて 3 種類の方式 (CLB 単位、列単位、行単位) のいずれかを用いて回避する手法が提案されている。

Wei-Je Huang(Stanford University) らによって 2001 年に FCCM で発表された [7] では、Xilinx のような部分再構成可能なデバイスをターゲットとし、物理故障を検出した部分を含む構成単位 (カラム) を避けて再構成する手法を提案している。

Daniel Mange(Swiss Federal Institute of Technology) らによって、2000 年に EH で発表された [8] では、自己複製可能デバイスを前提に、物理故障を検出した際空きセルに構成情報をコピーし代替処理を行わせることで物理的障害に対する代替処理を行う手法が提案されている。

John Emmert(University of Kentucky) らによって 2000 年に FCCM(IEEE Symposium on Field-Programmable Custom Computing Machines) で発表された [9] や、Miron Abramovici らによって 2001 年に EH で発表された [10] では、Self-Testing AREas(STARs) を FPGA 上で動作させることで、物理故障の検出とその位置の特定を行う手法を提案している。検査中にも検査部分以外は動作し、故障箇所が発見された場合でも他の部分は動作を続けることができる。

Steven K. Sinha(Carnegie Mellon University) らによって 2000 年に FCCM で発表された [11] では、パイプレンチ用に効率的な BIST(Built-in Self-Tests) アルゴリズムを開発し、制御ロジックをわずかに変えることで物理故障への対処を達成している。パイプレンチではアプリケーション動作中に BIST を行うことができる。

坪井秀幸(日本電信電話株式会社) らは、2003 年の RECONF において発表した [12] で、再構成可能部の試験機能と自己複製機能を組み合わせ、自己複製と試験を交互に行う手法を提案した。その結果、外部から全ての再構成可能部にテストパターンを入力する手法に比べ、試験時間を大幅に短縮できる。ただし、境界 LUT の試験が困難にならないように、境界に簡単な回路を追加している。このようにハードウェア上で実装することにより、試験コストの削減を確認している。

Nicholas J. Macias(Cell Matrix Corporation) などによって 2002 年に EH で発表された [13] では、自己再構成可能なデバイスである Cell Matrix 上に領域の検査を行いつつ自身の複製を繰り返す Supercell を構成し、物理故障の検査と回避を行う手法が提案されている。処理は駆動前に行われ、検査および複製によって Supercell が敷き詰められると、各 Supercell はゲートの割り当てとゲート間の配線を行う。

2.4 実行時の論理故障を扱うもの

2.2 節で紹介した [5] に加え、以下のような研究成果が発表さ

れている。

Wei-Je Huang(Stanford University) らは 2001 年に発表した [1] において、FPGA 内部でメモリとして構成された部分を動作中に修正することによって発生するメモリ内容の同一性の問題に対処するための手法を提案している。構成情報における論理故障の検出および修復の手法は、できるだけ FPGA の動作を妨げないように部分的に順に検査および修復する、というものである。将来は、この手法を動的再構成可能デバイスにも応用することを考える。

Andrzej Krasniewski(Warsaw University of Technology) らによって 2004 年に DFT で発表された [14] では、順序回路を対象として、レイテンシ無しで故障検出する手法の提案を行っている。ステートマシンの状態と入力の組み合わせには、無効なものが多く存在することを利用する。ハードウェアではなく構成情報の追加によって故障の検出を行い、それが可能であることを証明している。

これまで紹介した研究では、既存デバイスを運用する上での耐故障化、物理故障を扱った手法、静的診断や静的回復を行うものが多く、いずれも稼働中の論理故障に対しての完全な対処は難しい。特に、既存 FPGA などの再構成可能デバイスは、稼働前に書き換えることを主眼として開発されたものであるので、故障が発見されれば停止を余儀なくされ、その結果、復帰するまでに大きなオーバーヘッドが発生する。また、FPGA 内部に耐故障機能を構成する場合、その機能自体が故障しやすいという問題がある。そこで我々は、実行時に起こる論理故障に対して、動的に検出と回復を行う手法に関しての検討を行っていく。

3. 耐故障動的再構成可能アーキテクチャの検討

この章では、現在我々が考えている検討事項と、その前提事項について述べる。

再構成可能デバイスの機能は柔軟に変更可能だが、構成情報がメモリ素子に記憶されることから、ビット反転が起こる場合は処理対象のデータだけでなく構成情報にも発生する。その結果、間違った演算が行われたり、意図しない結果やデバイスの破壊を招くおそれもあるため、その検査や修正のための機構を考える必要がある。

第 2 章では、物理的な故障箇所を避けるものや、既存の FPGA の構成情報を多重化するものといった従来研究を紹介したが、我々は耐故障機能自体を固定ハードウェアで実装し、より高い信頼性を得ることを考える。さらに動的再構成可能デバイスに着目し、動作中に書き換えられる特長を生かして論理故障を克服することを考える。

故障への対処手順の概要を図 1 に示す。耐故障性を向上させるには、故障を検出および特定するために冗長性を高める必要がある。代表的には、同じ処理を複数回行うなどの時間的に冗長性を高める手法と、同じ処理を行う機能を複数用意する(多重化など)、検出や訂正のための専用の機能(誤り訂正符号など)を付加するなど、空間的に冗長性を高める手法がある。次

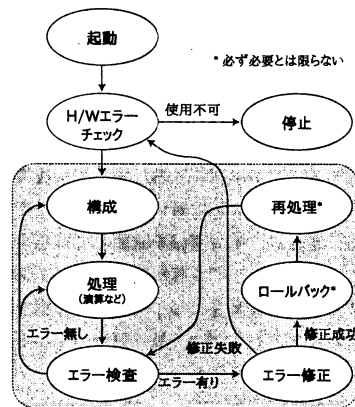


図 1 故障検出および修復処理の概要

節以降では、図 1 の網掛け部分に相当する論理故障への対処手法として、多重化手法および誤り訂正符号を用いる手法の基本的な検討を行う。

3.1 系を多重化する手法

入力データを複数の同じ構成の処理ユニットで処理する手法である。ここでは、図 2 のように 3 重化した場合について検討する。

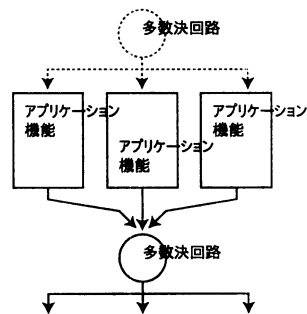


図 2 3重化システムの例

最大のねらいは、多重化と多数決を組み合わせることで、停止やロールバックを行わずに障害への対処を行うことである。システムは大きく分けて、3重化された処理ユニットと、その制御部分からなる。制御部分には、各処理ユニットからの出力の同一性を判定する多数決部と、故障が発生した場合の構成情報修復制御部が必要となる。

図 3 に、3重化における状態遷移図を示す。同じデータを複数の同じ処理ユニットで処理するので、正常に動作していれば出力されるデータは全て同一のはずである。全ての出力データが同一な状態が「正常」状態である。もし、出力されたデータに異なるものがある場合、いずれかの処理ユニットに障害が発生したことが判る。3つのうち1つだけ出力が異なる場合、出力が異なるものに故障があると考えられる。この場合は2重化で処理を行い、その間に故障箇所の修復が可能である。出力

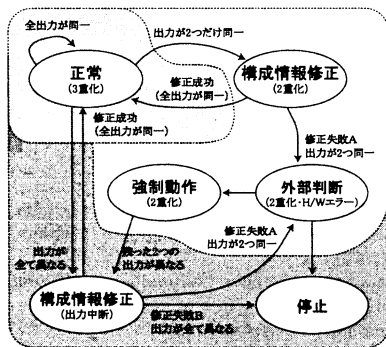


図3 3重化回路の状態遷移

が3つとも異なる場合は故障箇所の特定は不可能であり、処理を中断して修復する必要がある。このように、1箇所までの故障に関しては停止させずに動作させられるので、性能の低下は少ないと考えられる。しかし、処理ユニットなどがほぼ3倍になってしまうので、回路規模が非常に大きくなってしてしまうというデメリットが考えられる。

3.2 誤り訂正を用いる手法

構成情報に誤り訂正可能な符号を付加し、構成情報を検査および訂正することで論理故障に対する耐障害性を向上させる手法である。

このシステムでは、1度に全ての構成情報を検査できるとは限らないため、構成情報に論理故障が発生しても即座に検出されるとは限らない。そのため、論理故障が発生してから検出までに誤ったデータが出力されるおそれがあり、発生した場合は再演算処理が必要となる。そのために演算データを保持するためのバッファメモリが必要となる。そして、そのバッファ容量の制限から、一定時間（バッファが満たされるまで）以内に1度は検査が必要である。また、再構成されて全く新しい機能に

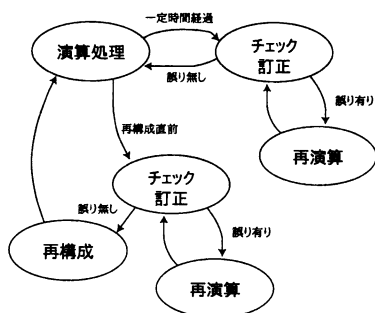


図4 誤り訂正を用いる回路の状態遷移

書き換えられると、その前の状態に論理故障があってもそれが検出されないので、再構成直前にもチェックする必要がある。これらのことから考えられる状態遷移図が、図4になり、エラー訂正の概要は図5のようになる。このシステムには、誤り訂正符号を含む構成情報メモリ、そこから誤りを検出して修正するための回路、再演算に必要なバッファメモリ、これらの制御を

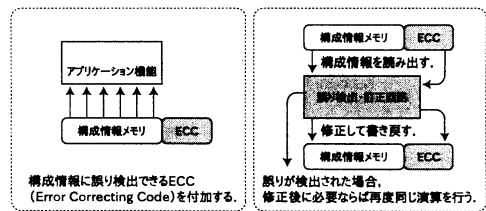


図5 誤り訂正を用いるシステムの例

する回路が必要となる。バッファメモリには、たとえばダブルバッファやデュアルポートメモリを用いることなどが考えられる。これら必要な回路から考えられるこのシステムのブロック図の概要を、図6に示す。この処理系全体の状態を管理、制御

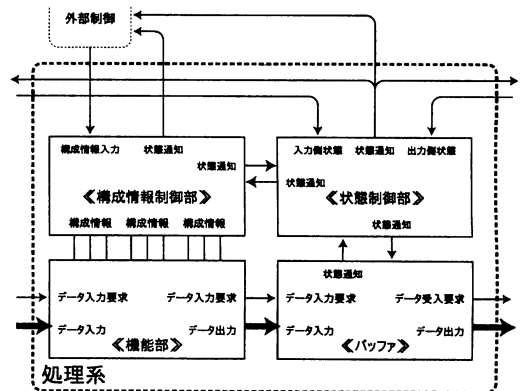


図6 誤り訂正を用いるシステムのブロック図（概略）

する部分が、状態管理部である。構成情報制御部は、状態管理部から受け取る状態に応じて図5に示す機能、すなわち構成情報メモリの誤り訂正処理および再演算が必要かどうかの判定を行う。バッファは、再演算に必要な演算データを保持し、状態管理部から受け取る状態に応じてデータ出力の制御も行う。機能部は、構成情報制御部にある構成情報メモリの内容に応じた演算機能を実現する。

このシステムでは、3重化の場合と比較すると回路規模の増加は少ないと考えられるが、物理故障の検知は難しいと考えられる。性能に関しては、論理故障が発生しない状態ではレイテンシが大きく、スループットの低下はあまり無いが、論理故障が発生した場合は、訂正や再演算のペナルティが大きく、スループットの低下が考えられる。また、バッファサイズが大きくなるにつれ、レイテンシは大きくなっていく。システムの制御に関しては、故障検出のタイミングが一定時間毎と再構成前であり、故障検出された場合には再演算が必要など、3重化よりも複雑である。特に、再演算の際には再度同じデータを受け取らなければならないので、故障が発生した処理系の状態と、その系と関わりのある処理系（およびバッファ）の状態とが矛盾を起こさない制御が必要である。

それらの問題を検討し、改良した状態遷移図を図7に示す。誤り検査のタイミングと再構成のタイミングを、いずれも一定

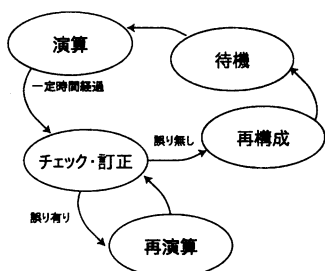


図 7 誤り訂正を用いる回路の状態遷移状態遷移図 2

時間毎に統一し、「待機」状態（全ての処理が完了する状態）を設置した。待機状態は、関わりのある処理系が全て待機状態になるまで待つ状態である。こうすることで、制御を簡略化し、再演算の際も確実に同じデータが得られるようにする。この状態遷移図から考えたシステムのブロック図の概要は、図 6 と同じである。

今後、以上のような検討をさらに詳細化していき、目標となる高信頼性動的再構成デバイスの実現を目指していく。

4. まとめと今後の指針

再構成可能デバイスは、その再構成能力への期待から、宇宙開発分野などから高い関心が寄せられている。しかし、宇宙線などの影響による論理反転が構成情報に起こるおそれがあり、それによる影響がどのようなものになるのか判らず、システムに致命的な損傷を与える可能性すらある、という問題がある。

本稿では、その問題の解決に向け、再構成可能デバイスにおける耐故障機能についての調査と、我々が考えている検討事項を述べた。従来の研究では、既存の FPGA を用いたものが多く見受けられ、その FPGA 内に耐故障機能を搭載したり、FPGA の機能を利用した手法などが提案されていた。我々はそれらに対し、動作中に書き換えが可能な動的再構成可能デバイスを用い、耐故障機能を固定ハードウェアとして組み込むことで、さらに高い耐障害性を実現することを提案した。そのために、冗長性を高める代表的な手法である、多重化を用いる手法とエラー訂正符号を用いる手法の検討を行った。

今後は、調査を継続していき、各手法に関する詳細な検討を重ねていき、論理故障に対して高い耐故障性を持った動的再構成可能デバイスを提案し、性能や回路規模などと信頼性とのトレードオフの評価を行っていく。そして、論理故障に加え、物理故障にも対応できるシステムを検討していき、総合的に高い信頼性を持った動的再構成可能デバイスを提案していく。

文 献

- [1] W.-J. Huang and E. J. McCluskey: "A Memory Coherence Technique for Online Transient Error Recovery of FPGA Configurations", Proc. of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, pp. 183-192 (2001).
- [2] J. Moreno, J. Madrenas, J. Cabestany, E. Canto, R. Kielbik, J. Faura and J. Insenser: "Realization of Self-Repairing and Evolvable Hardware Structures by Means of Implicit Self-

Configuration", Proc. of the First NASA/DoD Workshop on Evolvable Hardware, pp. 182-187 (1999).

- [3] T. Bartzick, J. K. M. Henze and K. Woska: "Design of a Fault Tolerant FPGA", Proceedings of the The Roadmap to Reconfigurable Computing, 10th International Workshop on Field-Programmable Logic and Applications, pp. 151-156 (2000).
- [4] A. H. Jackson, R. Canham and A. M. Tyrrell: "Robot Fault-Tolerance Using an Embryonic Array", Proc. of 2003 NASA/DoD Conference on Evolvable Hardware, pp. 101-110 (2003).
- [5] P. Lala and A. Walker: "An on-line reconfigurable FPGA architecture", Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, pp. 275-282 (2004).
- [6] P. Sundararajan and S. A. Guccione: "Run-Time defect tolerance using JBits", Proc. of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, pp. 193-198 (2001).
- [7] W.-J. Huang and E. J. McCluskey: "Column-Based Pre-compiled Configuration Techniques for FPGA Fault Tolerance", Proc. of the 2001 IEEE Symposium on Field-Programmable Custom Computing Machines (2001).
- [8] D. Mange, M. Sipper, A. Stauffer and G. Tempesti: "Toward Self-Repairing and Self-Replicating Hardware: The Embryonics Approach", Proc. of the Second NASA/DoD Workshop on Evolvable Hardware, pp. 205-214 (2000).
- [9] J. Emmert, C. Stroud, B. Skaggs and M. Abramovici: "Dynamic Fault Tolerance in FPGAs via Partial Reconfiguration", Proc. of the 2000 IEEE Symposium on Field-Programmable Custom Computing Machines (2000).
- [10] M. Abramovici, J. M. Emmert and C. E. Stroud: "Roving STARS: An Integrated Approach to On-Line Testing, Diagnosis, and Fault Tolerance for FPGAs in Adaptive Computing Systems", Proceedings of the Third NASA/DoD Workshop on Evolvable Hardware (EH.01) (2001).
- [11] S. K. Sinha, P. M. Kamarchik and S. C. Goldstein: "Tunable Fault Tolerance for Runtime Reconfigurable Architectures", Proc. of the 2000 IEEE Symposium on Field-Programmable Custom Computing Machines (2000).
- [12] 坪井, 塩澤, 小西, 中田, 伊藤, 名古屋: "Sea-of-LUTs 型動的再構成ハードウェアの試験方法", 第 1 回コンフィギャラブルシステム研究会, pp. 133-140 (2003).
- [13] N. J. Macias and L. J. K. Durbeck: "Self-Assembling Circuits with Autonomous Fault Handling", Proc. of 2002 NASA/DoD Conference on Evolvable Hardware, pp. 101-110 (2002).
- [14] A. Krasniewski: "Concurrent Error Detection in Sequential Circuits Implemented Using Embedded Memory of LUT-Based FPGAs", Proceedings of the 19th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, pp. 487-495 (2004).