Vol.2009-ARC-184 No.4
2009/8/4

耐永久故障FPGAアーキテクチャ

岡田崇志[†1]　喜多貴信[†1]
五島正裕[†1]　坂井修一[†1]

宇宙で使用される電子機器は，放射線によりシングル・イベント効果などの故障を引き起こしやすいため，高い放射線耐性が要求される．また修理，交換が困難なこともあり，自律的に故障から回復する機構が求められる．本稿では，通常用途のFPGA (Field-Programmable Gate Array) にわずかなハードウェアを追加するだけで，過渡故障，永久故障に対する耐性を付加する手法を提案する．本手法では，機能回復ロジックをFPGAのユーザロジック上に構成することで，コンフィギュレーションデータを計算するための追加ハードウェアの必要をなくす．この手法により生じる面積オーバヘッドは小さいため，通常用途と高信頼用途の両立が可能であると考えられる．

Fault-tolerant FPGA Arichitecture

Takashi Okada,[†1] Takanobu Kita,[†1]
Masahiro Goshima[†1] and Shuichi Sakai[†1]

Since electric devices for space applications are likely to experience radiation induced errors, such as the Single Event Effects, they must be designed to protect against faults. In addition, it is difficult to replace or fix faulty units in space. For that reason, autonomous detection and recovery machanism is required. This paper presents a fault-tolerant FPGA (Field-Programmable Gate Array) architecuture that requires little additional hardware. We incorporate an RM (Recovery Manager) into the FPGA 's user logic, which means no special hard-wired logic is required for calculating configuration data. The area overhead is so small in our apporach that we can devise FPGAs that are useful for both normal and critical operations.

†1 東京大学大学院情報理工学系研究科
Graduate School of Information Science and Technology, The University of Tokyo

1. はじめに

宇宙用途のLSIは，地上とは異なる，以下のような過酷な環境に曝される．

- 高いエネルギーの放射線が頻繁に降り注ぐため，この放射線の衝突により，電子部品には，メモリ情報反転（シングル・イベント効果）と放射線エネルギーの蓄積効果（トータル・ドーズ効果）が同時に発生する．
- 高真空環境であり，衛星内部機器であっても $-30°C$ から $+60°C$ の温度サイクルが加わり，機器の劣化を早める要因となる．

これらの複合的な環境要因のため，**過渡故障**（transient fault, soft fault）のみならず，**永久故障**（permanent fault, hard fault）の確率も跳ね上がる．

また，人工衛星等に搭載される宇宙用途のLSIに特徴的なこととして，交換が困難であることが挙げられる．航空用途を含む非宇宙用途のLSIの場合，故障を起こしたとしても，最悪そのLSIを交換すればシステムの運用を続けることができる．一方，宇宙用途では，交換が極めて困難であるため，1個のLSIの故障がシステム全体の永久的な運用停止につながり得る．

以上のような要因のため，宇宙用途のLSIには，地上用途に比べて，格段に高い信頼性が要求されるのである．

高い信頼性を確保するため，宇宙用途においては，米軍使用の高信頼性部品**MIL規格部品**が使用されてきた．しかし，MIL部品は，古い世代のプロセスで製造されているので民生品と比べ，性能面で劣る．そのうえ，高価，長納期であるというデメリットを持っている．

そこで本稿では，通常用途向けの**FPGA**（Field-Programmable Gate Array）をベースとして，十分に小さい追加ハードウェアで耐故障性を付加することを考える．このアプローチによれば，MIL部品などを使う際に生ずる問題は以下のように解決される：

(1) 通常FPGAを使用可能
耐故障性のための追加ハードウェアは十分に小さければ，通常のFPGAからのコストアップも最小限に抑えられる．したがって，通常用途向けに大量に製造し，安価に販売しながら，その全く同一のFPGAを宇宙用途向けに使用することが可能になる．

(2) 性能面での優位
民生品と同様の最先端のテクノロジを用いて製造される．したがって，FPGAであっても，MIL部品など，2〜3世代前のテクノロジで製造されるASICと比して十分な性能が提供できる．

一方で，通常用途向けの FPGA をベースとして，十分に小さい追加ハードウェアで耐故障性を付加するためには，以下のような点について考慮する必要がある．

まず，最先端の微細なテクノロジを使うため，放射線に対する感受性が高い．そのため，必然的に，故障の**回避**（avoidance）ではなく，故障の**検知**（detection）と**回復**（recovery）によって対処することになる．基本的には，**TMR**（Triple Modular Redundancy）などを用いて故障を検出し，**動的部分再構成**（Dynamic Partial Reconfiguration: **DPR**）によって回復を行うことになろう．

実際，類似の提案は多い[1)–3)]．しかし既存の研究では，DPR のための（再）構成情報を計算する**回復マネージャ**をハードワイアード・ロジックで構成する例がほとんどである．そのような方式では，回復マネージャが大きくなりがちで，追加ハードウェアを十分に小さくするという目的を達成できない．また，回復マネージャ自体が**単一故障点**（single point of failure）になってしまっている提案もある．

そこで本稿では，回復マネージャを，ハードワイアード・ロジックではなく，**ユーザ・ロジック**で構成する手法を提案する．回復マネージャ自体をユーザ・ロジックとして TMR で構成すれば，回復マネージャが単一故障点となることは避けられる．一方で，回復マネージャ自体をユーザ・ロジックとして TMR で構成するということは，回復マネージャを構成するロジック上に故障が検出された場合，回復マネージャが回復マネージャ自体の DPR を行うことを意味する．

構　　成

以下，第 2 章と第 3 章では，背景知識として，LSI にとっての宇宙環境と，提案のベースとなる FPGA とそれに対する故障耐性技術についてまとめる．第 4 章では，DPR を用いた故障耐性技術について既存のものを取り上げる．第 5 章では，提案手法について詳しく説明する．

2. 宇 宙 環 境

本章では，宇宙空間で起こりうる放射線エラーについて，具体的に説明し，その後，宇宙用途で用いられる MIL 部品について説明する．

2.1 放射線に起因する故障

衛星に搭載される機器は，高いエネルギーの放射線に曝される．このような放射線が原因で LSI に発生する故障は，大きく 2 種類に分けられる．一つは偶発的に起こる現象であり，**シングル・イベント効果**(Single Event Effect:SEE)[7)] と呼ぶ．もう一つは，長時間放射線に曝されることによって生じる経年現象であり，**トータル・ドーズ効果**(Total Ionizing Dose:TID) と呼ぶ．この節では SEE 及び TID について説明する．

- シングル・イベント効果
  放射線によって確率的に起こる故障をシングル・イベント効果と呼ぶ．主な現象としてシングル・イベント・アップセット（Single Event Upset:SEU）がある．これは，宇宙線と半導体メモリの構成物質であるシリコン原子核との反応によってメモリ内の情報が反転する現象である[4)]．これは，デバイスのハードウェアに永久的な損傷を与えるものではないため，過渡故障に分類される．検出した段階でデバイスをリセットしたり，正しい値を上書きすることで正常な動作に戻すことができる．
- トータル・ドーズ効果
  放射線によって生じる効果は，偶発的な現象だけでなくダメージの蓄積によって引き起こされるものもある．これは，トータル・ドーズ効果と呼ばれ，放射線に長時間さらされることによって，トランジスタの動作速度の低下，リーク電流の増加などが引き起こされる．最終的には永久故障につながる．

2.2 MIL 部品

宇宙空間での，放射線エラーに対し，高い耐性を確保するため，宇宙用途においては，従来，主に米軍仕様の高信頼性部品 **MIL 規格部品**が用いられてきた．たとえば，次世代型無人宇宙実験システム（USERS）衛星バス[⋆1]では，使用されている MIL 部品数は約 37,900 点で，その購入費用は全体価格の 22% を占める．また，電子部品費用の 87% がマイクロプロセッサ，メモリ，ゲート・アレイ等の高機能な電子部品で占められている[8)]．

しかし MIL 部品は，信頼性こそ高いものの，民生部品と比較すると以下のようなディメリットがある：

- 低性能，高価，長納期である．
  - 半導体の微細化にともない，LSI は放射線などの影響を受けやすくなるため，民生品より 2〜3 世代古いテクノロジで製造される．そのため，チップ面積は大きく，動作周波数は低く，消費電力は大きい．
  - 製造量/出荷量が少ないため，高価かつ長納期である．
- 調達が困難になる可能性がある．製造量が少なく採算が合わない等の理由から，部品メーカの MIL 部品生産からの撤退が顕著となっており，今後，必要な部品が調達でき

⋆1 衛星バス：全ての衛星に備わっている共通的部分で，通信系，姿勢制御系，太陽電池パドル系，電源系等からなる

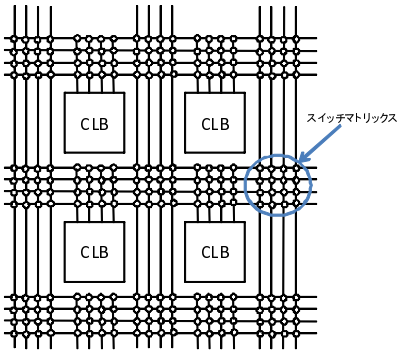

図 1 FPGA の構造
Fig. 1 FPGA structure

なくなる可能性も決して低くはない．

3. FPGA と故障耐性技術

本節では，まず FPGA の構造と特徴を説明した後，FPGA において用いられる故障耐性技術について述べる．

3.1 FPGA の構造

FPGA の内部仕様は各メーカによって異なるが，基本的な原理は同じであると考えてよい．本稿では，Xilinx 社の FPGA を例として説明する．

図 1 に示すように，FPGA 内部には CLB（Configurable Logic Block）と呼ばれるブロックが格子状に並び、それらが相互にスイッチマトリックスで接続された構造となっている．CLB は FPGA の回路を構成する基本単位である．

組み合わせ回路は CLB 内にある LUT（Look Up Table）によって実現される．LUT とは，SRAM で構成されたテーブルであり，任意の真理値表を格納できる．入力値をアドレスとして，このテーブルを参照し，対応した値を出力することで，組み合わせ回路を実現できる．図 2 は二入力 LUT で AND 回路を構成した例である．

CLB を用いて，小規模な組み合わせ回路，シフトレジスタ，RAM などを実現できる．設計者が回路を実装する場合，論理回路を CLB にマッピングし，それらをつなぐ配線を決定する必要がある．通常このような作業は CAD によって自動化されている．

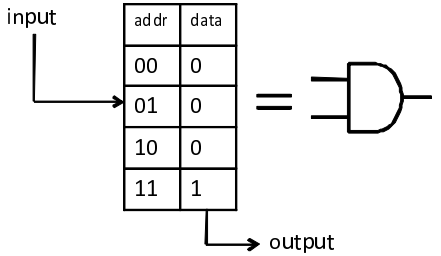

図 2 二入力 LUT による AND ゲート
Fig. 2 AND gate implementation with a two-input LUT

3.2 コンフィギュレーション

FPGA に回路を構成することをコンフィギュレーションと呼ぶ．FPGA の回路の，プログラム可能な機能は全て，デバイス内にあるコンフィギュレーションメモリに書き込まれたコンフィギュレーションデータによって決定する．コンフィギュレーションデータには，信号のルーティング情報，LUT の内容，入出力端子の電圧などの情報が含まれており，通常は FPGA ベンダーが提供するツールによって，HDL[⋆1]で書かれたソースコードを元に自動的に生成される．

FPGA に回路を書き込むには，コンフィギュレーションメモリのコントローラに対するコマンドが埋め込まれたコンフィギュレーションデータ（ビットストリーム）を指定のポートにダウンロードすればよい．

3.3 動的部分再構成

FPGA に生じた故障から回復するためには，回路を再構成する必要がある．ここでは，その仕組みについて説明する．

コンフィギュレーションメモリは，フレームという単位で分割されている．このフレームという単位は，一度の操作で書き換え可能な最小単位であり，フレームアドレスと呼ばれるアドレスで特定のフレームを指定し，部分的に一つのフレームのみ，データを書き換えることが可能である．Viretx-4（XC4VSX35）を例にあげると，全コンフィギュレーション

⋆1 ハードウェア記述言語．集積回路を設計するために用いられるプログラム言語

データは約 1.7MByte，フレームの大きさは 164Byte であり，一つのデバイス内に 10410 個のフレームが存在する[6]．

Xilinx 社の Virtex シリーズでは，動作中にコンフィギュレーションデータの一部を書き換える方法をサポートしている．あるフレームに対してコンフィギュレーションデータの書き換えを行った場合，他のフレームに書き込まれたデータには影響を与えないため，残りの回路の動作を継続したまま一部分のみを再構成することが可能である．これを**動的部分再構成**と呼び，本稿ではこの手法を自律的な故障修復のメカニズムの基本とする．

3.4 TMR

一般的によく用いられる故障耐性技術の一つ，**TMR** について説明する．TMR は，回路を三重化して冗長性を持たせ，故障耐性を付加する手法である．この手法は，宇宙用途の電子機器において故障耐性を得るために，よく用いられる．TMR を構成するには，まず，同一の回路を三つ用意し，同一のタイミングで同じ処理をさせる．三つの回路の出力は多数決回路（voting logic）によって比較され，最終的な結果として，多数派の値が選択される仕組みになっている．

この手法を用いることによって，三つの回路のうち，仮に一つに故障が発生したとしても，他の二つが正常な値を出力し続ける限り，正しい値が選択される．したがって，単一の回路に発生した故障はマスクされ，全体として回路は正常な動作を継続することができる．

また，TMR を用いることによって故障個所を特定することができる．図 3 のように三つの出力の値を比較する回路を付加することで，どのモジュールに故障が発生したかを知ることができる．

TMR は，面積オーバーヘッドが大きいが，過渡故障，永久故障が，どのような部位に発生しても検出可能である．FPGA は，論理ゲートに相当する部分が LUT，すなわち SRAM で構成されているので，制御部を含め，あらゆる部分でシングル・イベント・アップセットが発生する恐れがある．このような故障をリアルタイムで検知するためには，TMR が有効である．

3.5 Scrubbing

シングル・イベント・アップセット対策として，**Scrubbing**[5] という手法がある．この手法では，エラーの発生の有無にかかわらず，定期的に，外部メモリ保存してあるコンフィギュレーションデータで，デバイス全体を上書きする．コンフィギュレーションメモリに SEU が生じても，定期的にデータを上書きすることで訂正される．この手法は，ハードウェアをほとんど追加せずに実装できるが，コンフィギュレーションデータを上書きするだけでは，永久故障から回復することはできない．

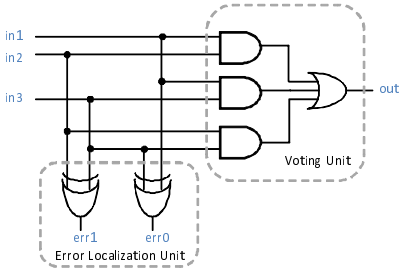

図 3 TMR による故障個所検出法
Fig. 3 Fault detection through TMR

4. 関 連 研 究

故障を検知し，再構成によって永久故障から回復する手法がいくつか提案されている．

2) で提案されている手法は，ロジック部や，配線部に発生した故障を，動的に再構成することで回復する手法である．この手法では，故障が発生した部位に応じて，複数の異なる回復アプローチをとる．ワーストケースでは，故障が発生したブロックを正常なブロックに置き換える操作が必要となるため，この際，再配線するために外部に用意されたプロセッサに頼る必要がある．

3) で，提案されている手法は，ソフトエラー，ハードエラーそれぞれに対して，別のアプローチで故障回復する．ソフトエラーに対しては，コンフィギュレーションデータの上書きで対応し，ハードエラーに対しては，故障部をバイパスするように新しく回路を再構成することで回復する．この手法では，故障耐性を持ったマイクロコントローラの存在を前提としており，故障検出，回復はそのマイクロコントローラによって行われる．

一方で，回路診断の分野でも，再構成を利用したものがある．1) で提案されている Roving STAR と呼ばれる手法がその一つである．STAR とは Self-Testing AREa を表わし，ワーキングエリアとは独立した，診断中の領域のことを指す．

まず，回路の一部分に，予備の領域を用意しておき，そこに STAR を構成する．STAR は，その領域の診断を終了すると，隣の領域に移動する．これを繰り返し，STAR が全領域をくまなく巡回することで故障を検出する．診断中の領域は，本来の機能が失われる．しかし，STAR が自身の領域を移動する際に，移動先の回路の機能を現在の領域にコピーし，

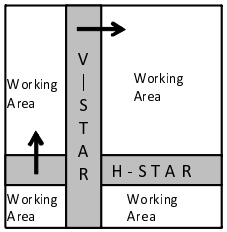

図 4 STAR
Fig. 4 STAR

再配線するので，動作中の回路に影響を与えずに全領域の診断を行うことができる．図 4 に示すように，STAR は回路を水平に横切る H-STAR と垂直に移動する V-STAR の二種類があり，双方が個別に診断を行う．この手法も，回路診断，再構成の機能は故障耐性を前提としたマイクロコントローラが受け持つ．

これらの方法では，再配置，再配線の際に計算が必要であるため，プロセッサなどの計算資源を追加することが必要である．また，追加したハードウェアの故障耐性は前提条件となっており，十分に議論されているとはいえない．しかし，システム全体の故障耐性を議論する場合，追加したハードウェアについても，十分な故障耐性を保障することは必要不可欠である．5 章で，このような問題に対するアプローチについて説明する．

5. 提 案 手 法

5.1 回復マネージャ

再構成を用いた故障回復では，故障した回路を物理的に離れた別の場所に再構成する必要がある．その際に，新たに配置配線を行う必要があり，これには比較的大きな計算コストがかかるため，何らかの計算資源が必要である．従来の手法には，そのような計算のための専用ロジック（ここでは回復マネージャと呼ぶ）を，ハードワイアード・ロジックで追加することを想定したものが多い．回復マネージャは，一度故障すればシステム全体の故障回復能力が失われることになるため，他の回路と同程度の故障耐性を備えていなければならない．そこで，回復マネージャ自体に故障耐性を付加しようとすると，追加するハードウェア量がさらに大きくなってしまうという問題がある．

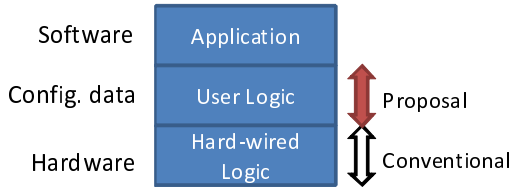

図 5 回復マネージャの実装領域
Fig. 5 Recovery Manager Implementation

本手法では，回復マネージャを FPGA のユーザ・ロジック領域[⋆1]に他の回路と同様に実装する（図 5）．さらに，回復マネージャ自身に生じた故障を自分自身で検知，修復できる仕組みを導入する．この方法によって以下のようなメリットが生じ，先述した問題を解決できる．

- 回復マネージャ自身にも，他のユーザ・ロジックと同様の高い故障耐性を確保することができる．特に，FPGA の再構成可能な特徴を利用するので，ハードワイアード・ロジックでは実現できない柔軟な故障回復が可能である．
- 追加すべきハードウェアを小さく抑えることができる．回復マネージャはユーザ・ロジックとして，FPGA 上の，ある領域に構成される．それは，ハードワイアード・ロジックとは独立したものであるから，通常用途の FPGA として使用する際は回復マネージャを構成せず，その領域を自由な用途に割り当てることができる．

5.2 全 体 構 造

提案する手法の概略図を図 6 に示す．ユーザ・ロジック領域全体はタイルと呼ばれる単位で分割されており，全てのタイルが再構成ネットワークで接続された構造となっている．それぞれのタイルには，再構成ネットワークにアクセスするために，ネットワーク IF が 1 つずつ付属している．タイルの内部には，CLB が数百個程度含まれており，同一の機能を持つ CLB を三つ用意することで TMR を構成している．タイル間，CLB 間での，信号のやり取りは FPGA 上の通常の配線が使用されるが，故障情報，及び再構成データの伝達に限り，特別に用意されたネットワーク IF を通して再構成ネットワーク上で伝達される．

5.3 故 障 検 出

本手法では，TMR を用いて故障を検出するため，システムの正常な動作を妨げることな

⋆1 FPGA において，ユーザーが使用するプログラム可能な領域．

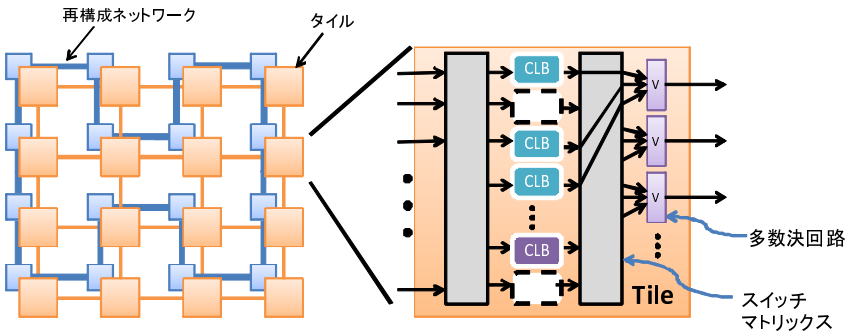

図 6 全体構成
Fig. 6 Whole structure

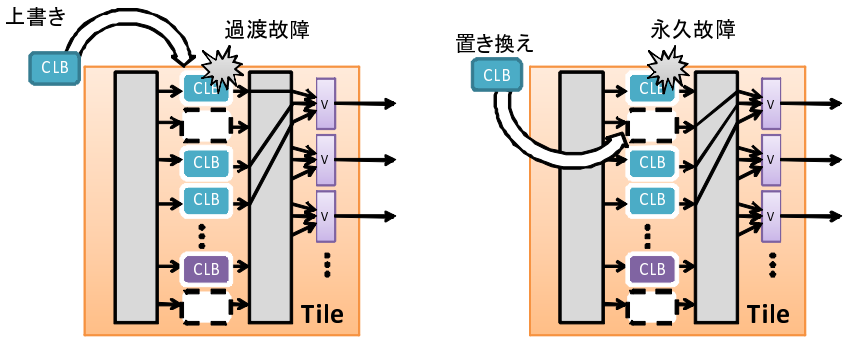

図 7 故障回復
Fig. 7 Fault Recovery

く，リアルタイムに故障検出，修復を行える．ここでは，TMR を用いて，故障けを検出する仕組みについて説明する．

デバイスのユーザ・ロジック領域はタイルという単位で分割されており，タイルの内部には数百個程度の CLB が含まれている（図 6）．全てのタイル内の CLB は三重化されている．すなわち，タイル中にある同一機能を持つ三つの CLB を用いて TMR が構成されている．CLB の出力は，ハードワイアード・ロジックで構成された多数決回路へ入力される．この際に，検出された故障信号は，ネットワーク IF を介して，再構成ネットワークに流され，回復マネージャへと伝達される．TMR を実現するためには，多数決回路の追加が必要であるが，これは図 3 に示すように，わずか 6 ゲートで実装可能である．TMR 一つ当たりおよそ 1000 ゲート程度の規模であると考えると，全体の面積に対してほぼ無視できる量のハードウェアである．

5.4 故 障 回 復

TMR は単一の故障をマスクすることができるが，一度故障が発生すると TMR の持つ冗長性は失われ，さらに別のモジュールに発生した故障に関しては，マスクされない．つまり，時間とともにシステム全体がダウンする確率が増加していくことになり，これでは，宇宙環境での動作を想定した場合，十分な故障耐性があるとは言えない．そこで，本手法では，故障を検出した際に，故障箇所を動的に再構成し，回復するアプローチをとる．故障検出は CLB 単位なので，この単位での再構成が可能である．過渡故障が発生した場合と永久故障が発生した場合の回復方法について，概念図を図 7 に示し，それぞれについて次に説明する．

なお，回復マネージャを含め，全てのロジックは三重化されており，故障が発生した際もシステム全体はオペレーションを継続できる．

過渡故障の場合 TMR を構成する CLB の一つに過渡故障が発生した場合，回復マネージャが故障した CLB のコンフィギュレーションデータを再計算し動的に再構成する．故障が起こった CLB を，元の正しい CLB で上書きすることで過渡故障から回復する．

永久故障の場合 故障が発生した際に再構成を何度か繰り返しても，回復しない場合は永久故障と判断する．永久故障が発生した場合には，元の CLB の複製を，タイル内の別の位置にある予備の CLB 上に構成し，再配線する．

5.5 再構成ネットワーク

自律的な故障回復を実現するために，中心的な役割を担うのが，本節で説明する再構成ネットワークである．全てのタイルは，再構成ネットワークのノードに相当し，ネットワーク IF を通して，アクセスできる仕組みになっている．再構成ネットワークは，大きく二つの機能を持つ．

一つは，故障情報とコンフィギュレーションデータの伝達である．タイル内の多数決回路から送信された故障検出信号，及び，再構成に必要なコンフィギュレーションデータは，ネットワーク IF を介して再構成ネットワーク上に，パケットの形で送信される．もう一つは，コンフィギュレーションメモリへのアクセスである．ネットワーク IF は，自身のタイルのコンフィギュレーションメモリに対する読み書きの機能を持っており，ネットワークを

通じて受信したコンフィギュレーションデータを用いてタイルを再構成することができる．

再構成ネットワークは，故障情報と，再計算されたコンフィギュレーションデータを伝達するために，特別に追加するネットワークである．単純に全てのタイル同士が接続するメッシュネットワークを実装すると，ルーティングを行うための，複雑な追加ネットワーク IF が必要となる．そこで，本手法では，追加ハードウェアを最小化するため，図 6 に示すように，トークンリングネットワークを用いた．これにより，ルーティングなどの必要がなくなり，ルータなどを追加せずに実現できる．故障回復の処理は，複数組のタイルが同時に通信を行う必要性がないため，トークンリングネットワークであっても十分に機能する．

さらに，このネットワークは十分に小さいハードウェアで実装可能である．なぜなら，このネットワークは故障回復の際にしか使われない専用ネットワークであるため，ある程度通信に時間がかかっても許容できる．そのため，1bit 幅の配線で送受信するシリアル通信を採用することが可能で，ハードウェア量を小さく抑えることができる．このようにシンプルなネットワークを採用することによって，ネットワーク IF は，タイル全体の回路規模に対し，十分小さい規模のハードワイアード・ロジックで実装することが可能であり，粗いプロセスで製作することができるので，この部分に故障が発生する確率は無視できるものとする．

5.6 故障回復の流れ

故障検知し，自律的に回復するまでの流れを説明する．

(1) ネットワーク IF が TMR の多数決回路からの信号を観測し，故障を検知する．
(2) 故障信号は，ネットワーク IF によってパケットの形に整形され，再構成ネットワークに流される．この際，リードバック機能により，正常な CLB も含めた故障タイル全体のコンフィギュレーションデータを読み出し，その情報もパケットに含める．
(3) このパケットを，別のタイルに実装されている回復マネージャが受け取る．
(4) 回復マネージャはリードバックされたコンフィギュレーションデータの中に含まれる正常な CLB の情報をもとに，新たに正しいコンフィギュレーションデータを計算し，これを含むパケットを，故障したタイル宛てに送信する．
(5) パケットを受け取った故障タイルのネットワーク IF は自身の CLB を再構成する．
(6) 故障信号が発生しなくなった場合は，この故障は過渡故障であったと判断しそこで一連の故障回復プロセスを終了する．
(7) もし，新しく回路を再構成しても故障信号が継続した場合，回復マネージャは永久故障と判断し予備の CLB に回路を移し替えるため，コンフィギュレーションデータを再計算し，対象のタイルにパケットの形で送信する．

以上のような流れで故障から回復する．

全ての回路は TMR で実装されているので故障回復の途中であってもシステム全体としては正常に動作し続けている．

回復マネージャ自体も，他のユーザ・ロジックと同じようにタイルの形で実装されており，ネットワーク IF を通じてネットワークにアクセスできる．仮に回復マネージャに，故障が発生しても，TMR で実装されているため，回復マネージャは動作を継続することができ，自身のネットワーク IF に対して，自分自身を再構成するように指示することができる．このように，故障回復のネットワークを導入することで，回復マネージャ自体も，再構成の対象とすることができ，単一故障点となることを回避している．

5.7 通常用途としての使用

以上で述べた故障回復の仕組みは，高い信頼性が求められる用途にのみ使用すればよい機能である．通常用途に使う場合は，ユーザ・ロジック領域に回復マネージャを構成する必要はない．また，ハードワイアード・ロジックで追加された，ネットワーク IF，再構成ネットワー及び多数決回路は無効化し，使用しない．これらの追加ハードウェアは，十分に小さく抑えられており，通常の用途の妨げになることはない．そのため，高信頼用途と通常用途の両方の用途で使用することが可能である．

6. ま と め

本稿では自律的に故障を検知し，修復する機能を持った FPGA のアーキテクチャを提案した．提案した手法では，回復マネージャをユーザ・ロジック領域に実装し，他の回路と同様に故障検出，回復の対象とした．このことによって，従来の手法では大きくなりがちであった，追加ハードウェアをほとんど必要とせず，故障耐性を持つ FPGA を構成できることを示した．同時に，本手法では，回復マネージャ自体の信頼性が確保できるため，システム全体の高い故障耐性を得ることができる．

通常用途の FPGA に対してハードワイアード・ロジックの追加は，最小限に抑えられるので，故障耐性を持つ宇宙用途 FPGA を，そのまま通常用途 FPGA として安価に大量に販売できる．(宇宙用途にも使用されているとなれば，通常用途に使用する際にも，製品の魅力の向上にもつながるであろう．) 結果として，宇宙開発のコストを削減できる．

提案した手法の実効性を確かめるため，評価用の FPGA ボード（図 8）を用意し，システムの論理設計を行ったが，実機での動作確認は行っていない．今後は，実機での動作確認，オーバーヘッドの評価などをしていきたい．

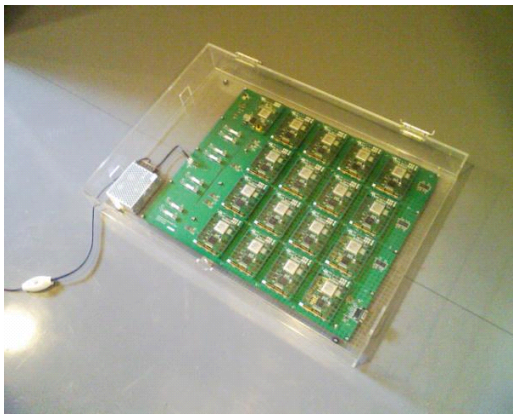

図 8 評価用ボード
Fig. 8 Testbed

参 考 文 献

1) Abramovici, M., Stroud, C., Hamilton, C., Wijesuriya, S. and Verma, V.: Using Roving STARs for On-Line Testing and Diagnosis of FPGAs in Fault-Tolerant Applications, *Test Conference, International*, Vol.0, p.973 (1999).
2) Lakamraju, V. and Tessier, R.: Tolerating Operational Faults in Cluster-based FPGAs, *in 8th International ACM/SIGDA Symposium on Field Programmable Gate Arrays*, pp.187–194 (2000).
3) Li, Y., Li, D. and Wang, Z.: A new approach to detect-mitigate-correct radiation-induced faults for SRAM-based FPGAs in aerospace application, *National Aerospace and Electronics Conference, 2000. NAECON 2000. Proceedings of the IEEE 2000*, pp.588–594 (2000).
4) Mukherjee, S., Emer, J. and Reinhardt, S.: The soft error problem: an architectural perspective, *High-Performance Computer Architecture, 2005. HPCA-11. 11th International Symposium on*, pp.243–247 (2005).
5) Xilinx: *Correcting Single-Event Upsets Through Virtex Partial Configuration* (2000). Xilinx Application Note 216.
6) Xilinx: *Viretx-4 FPGA Configuration User Guide* (2008).
7) YAHAGI, Y., IBE, E., YAMAGUCHI, H., KAMEYAMA, H., SAITO, Y., AKIOKA, T., YAMAMOTO, S., HIDAKA, M. and SAITO, A.: Evaluation of Terrestrial-Neutron Induced Single Event of Memory Devices : An Outlook for Logic Devices, *IEICE technical report. Dependable computing*, Vol.106, No.198, pp. 1–6 (20060725).
8) 独立行政法人新エネルギー・産業技術総合開発機構 (NEDO) 機械システム技術開発：「宇宙等極限環境における電子部品等の利用に関する研究開発プロジェクト」事業原簿 (2004).