電気学会論文誌 D（産業応用部門誌）
IEEJ Transactions on Industry Applications
Vol.139 No.2 pp.187–192　DOI: 10.1541/ieejias.139.187

論　文

部分再構成を用いたプロセッサの耐故障化手法に関する検討

非会員　荻堂　盛也*　正　員　山田　親稔*　非会員　宮城　桂*
正　員　市川　周一**　非会員　藤枝　直輝**

Research on Fault Tolerant Processor using Dynamic Reconfiguration

Seiya Ogido*, Non-member, Chikatoshi Yamada*, Member, Kei Miyagi*, Non-member, Shuichi Ichikawa**, Member, Naoki Fujieda**, Non-member

（2018年5月16日受付，2018年10月8日再受付）

In this paper, we propose a reconfigurable fault tolerant architecture that can recover from failure status with spare space. Recently, progress in semiconductor technology has been remarkable due to microfabrication of devices. The semiconductor technique plays an important role in artificial satellites and aircraft. Furthermore, it has guaranteed the reliability of the circuits by the multiplexing structure. However, in embedded systems, space-saving is regarded to be as important as reliability. In the traditional approach, the area overhead tends to become large. Reconfigurable fault tolerance can achieve high area efficiency. In this research, we aim to improve the reliability and area efficiency for a single stuck-at fault of the processor. In this article, we reproduce the proposed method using Tcl script and proposed standalone fault tolerant operation using embedded Linux.

キーワード：FPGA，動的部分再構成，フォールトトレラント，単一縮退故障
Keywords: FPGA，dynamic partial reconfiguration，fault tolerant，single event burnout

1. はじめに

半導体技術の進歩により，回路はより微細・集積化が進んでいる。打ち上げてからの修理が困難な人工衛星や，計器類に高い信頼性が求められる航空機では，半導体技術における信頼性は古くから重要な問題となっている。従来では，回路の重要な部分に多重化を施すなどのアプローチにより回路の信頼性を保証する試みがなされてきた[(1)(2)]。しかしながら，組み込みシステムに用いられるプロセッサは，信頼性と共に高い面積効率が要求される場合がある。こうした場合に，従来の面積冗長化回路は大きな面積オーバーヘッドを抱えてしまうことになる[(1)(2)]。近年では，回路をFPGA（Field Programmable Gate Array）などの再構成型デバイスに記述し，故障の度に再構成を行うことで故障状態から回復する再構成型耐故障アーキテクチャが注目されている。このアーキテクチャは多重化回路ほどのスループットは期待できないものの，デバイス上の余白領域に故障した箇所と同等の回路パターンを書き込むため，非常に高い面積効率を発揮することができる。

本研究では，プロセッサにおける単一縮退故障に対して，より容易かつ堅牢な再構成型耐故障アーキテクチャの実現を目的として，回路の動作を止めること無く信頼性を保証する事ができるDPR（Dynamic Partial Reconfiguration）と，コンフィグレーションメモリの信頼性を確保するScrubbingなどの技術を組み合わせた再構型耐故障アーキテクチャの検討を行った。ただし，本稿では故障の対象として単一縮退故障だけを扱うこととする。

以下では，先行研究[(3)~(5)]で提案されているFT-FPGAとその故障検出方法について説明し，提案するアーキテクチャの概要と，その動作の要となるDPR回路[(6)(7)]を用いたタイルの実装，さらに提案手法を自律的に動作させる手法について説明する。

2. 関連研究

〈2·1〉 タイルフォールトトレラント　再構成型耐故障アーキテクチャの代表的な例として，タイルフォールト

* 沖縄工業高等専門学校
〒905-2192　沖縄県名護市辺野古 905
National Institute of Technology, Okinawa College
905, Henoko, Nago, Okinawa 905-292, Japan
** 豊橋技術科学大学
〒441-8580　愛知県豊橋市天伯町雲雀ヶ丘 1-1
Toyohashi University of Technology
1-1, Hibarigaoka, Tempaku-cho, Toyohashi, Aichi 441-8580, Japan

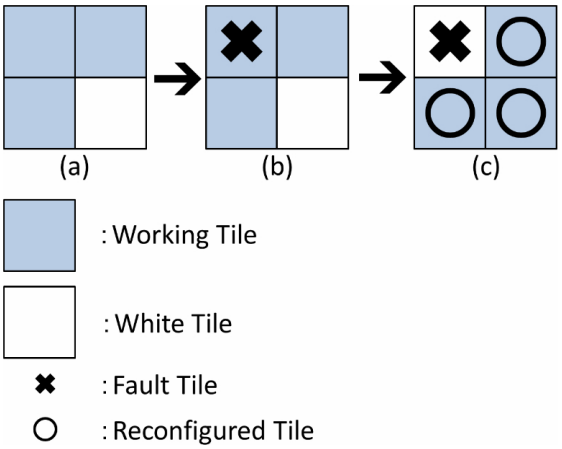


Fig. 1. A schematic diagram of tile fault tolerant.

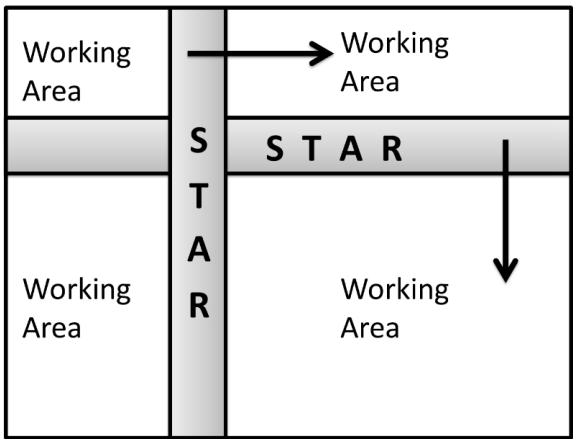


Fig. 2. Roving STAR.

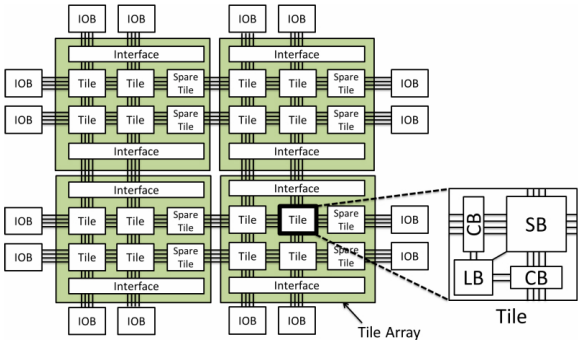


Fig. 3. A schematic diagram of FT-FPGA.

トレラント[8][9]について説明する。この手法は，FPGA 内をタイルと呼ばれるユニットに分割した構造になっている。FPGA 上に作成される回路はタイルの集合体として表現され，回路に故障が発生した場合は余白のタイルに故障した箇所と同等の回路を記述し，再構成することで故障から復帰する。Fig. 1 は，再構成により FPGA に書き込まれた回路機能が故障状態から回復するようすを表したものである。この手法はタイルフォールトトレラントと呼ばれており，追加のリソースを必要とせず，面積効率に優れているのが特徴である。問題点として，再構成の度にタイル同士を結線する配線のパターンが煩雑になっていき，配線遅延が増大することによる処理能力の低下が挙げられる。

〈2･2〉 Roving STAR タイルを用いた回路の検査手法について，代表的なものとして先行研究[10]で提案されている Roving STAR（Self-Testing ARea）について説明する。Roving STAR では，回路の余白の領域を利用し，そこに検査用の回路（STAR）を構成する。具体的には TPG（Test Pattern Generator）と ORA（Output Response Analyzer）の 2 種類の回路であり，TPG により生成されたテストパターンを対象の回路に流し，そこから得られた出力を ORA が受けとり，故障の有無を確認する。検査を終えると，隣の領域に検査用回路を新たに書き込み移動する。これを繰り返しながら巡回することで，回路全体から故障を検出する。Roving STAR では，検査を行なっている領域は通常の回路として機能しないが，検査先の領域の回路機能を現在の領域にコピーし，再構成することで全体の動作に影響を与えずに検査を行うことができる。この手法の短所は，タイルフォールトトレラントと同様に，再構成のために計算時間を必要とすることである。Fig. 2 に Roving STAR の概略図を示す。

〈2･3〉 FT-FPGA（Fault Tolerant Field Programmable Gate Array） 本研究で検討する耐故障アーキテクチャでは，単独で故障箇所の検出と回復動作を合わせて実行できるものを想定している。先行研究[3]~[5]で提案されている FT-FPGA（Fig. 3）は，独自のハードワイヤードロジックを追加することで，タイルフォールトトレラントと故障箇所の検出を両立させている。

FT-FPGA は，通常時に使用するノーマルタイル，故障時に回路を退避させるスペアタイル，退避後のタイルの接続補正を行うインタフェースから構成されている。タイルは，さらに LB（Logic Block），SB（Switch Box），CB（Connection Block）に分けることができる。FT-FPGA では複数のノーマルタイル，スペアタイル，インタフェースをまとめて 1 つのグループとみなしており，これを TA（Tile Array）と呼ぶ。また，TA 内に実装されたノーマルタイルの個数は TA サイズと呼ぶ。例を挙げると，ノーマルタイルの個数がタテヨコ n × n の場合は，TAn と表す。故障に対する検査・回避は TA 単位で行うため，Fig. 3 に示したアーキテクチャでは，4 つの TA が個別に故障に対して動作することができることになる。

次に，FT-FPGA の故障状態からの回復方法について説明する。故障を含む TA は，構成データを故障のないタイルへ移動させ，データの移動に伴うタイルの接続のズレを補正する。これにより，TA 内の構成データは故障箇所から右へシフトし，Fig. 4 の (a) は (b) へと変化する。ノーマルタイルとスペアタイルは構成データを共有できるようになっており，故障が発生した場合はセレクタ信号を切り替えるだけで構成データの移動を完了することができる。次に FT-FPGA による故障検出動作について説明する。Fig. 5

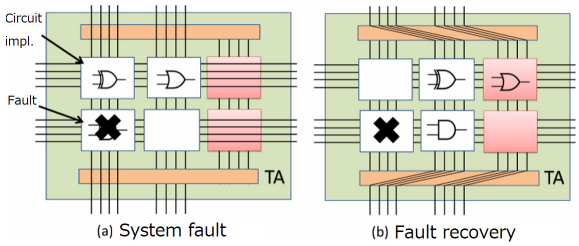


Fig. 4. Fault Recovery Operations.

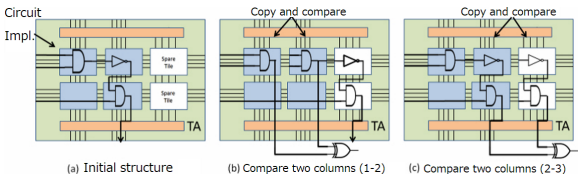


Fig. 5. Operation of Tile Array.

は，アレイサイズ TA2 の TA 内の検査動作のようすを表している。TA2 では，スペアタイルの一列分を含めて 2 行 3 列の形でタイルが並んでいる。左端の列から 1 列目と数えて検査を進める場合，Fig. 5 の (b) のように 1 列目のパターンを 2 列目に書き込み，1 列目と 2 列目で DMR（Double Modular Redundancy）を形成した上で 2 つの回路の出力を比較し，検査を行う。複製先の 2 列目のパターンは，スペアタイルである 3 列目に移動させる。1・2 列目の比較を終えると，Fig. 5 の (c) のように 3 列目のパターンを 2 列目に書き込むことで DMR を作成し，比較動作を行う。回路が実行動作中であっても検査は実行可能であるため，この処理は故障が検出されるまで繰り返される。先行研究 [3]~[5] では，回路の復帰・検出動作の実現のためにハードワイヤードロジックを部分的に用いている。アーキテクチャの容易な実装を実現するためには，FT-FPGA で実装されている機能を全てユーザーロジックで実現しなければならない。

〈2・4〉 DPR（Dynamic Partial Reconfiguration）

本研究において，再構成機構のベースとして検討した DPR [6][7] について説明する。再構成可能デバイスである FPGA は通常，回路デザインを上書きする際に，動作中の回路機能を全て停止させたうえで，ビットストリームを読み込み，再構成・再配線を行わなければならない。

しかし，Xilinx 社のサポートしている DPR（Dynamic Partial Reconfiguration）では，FPGA の領域をパーティションで区切り，フルビットファイルで FPGA をコンフィギュレーションした後に，パーティション内にパーシャルビットファイルを読み込むことにより，他の領域の回路動作に影響を与えずに再構成を行うことができる。DPR の特徴として，パーティションの設定による FPGA 領域のオーバーヘッドがない事が挙げられる。また，DPR の設計にはボトムアップ型合成と呼ばれる手法が用いられる。これは，結線情報や領域設定などのスタティックな部分と，再構成部分におけるダイナミックな回路デザインを階層に分けて設計するものである。スタティックデザインを上層，ダイナミックデザインを下層とし，下層で設計したデザインを上層で全て組み上げて，パーシャル領域の境界を示すパーティションピンの設定や，回路同士の結線情報を上層に落としこむことで設計する。ボトムアップ型合成の概略図を Fig. 6 に示す。

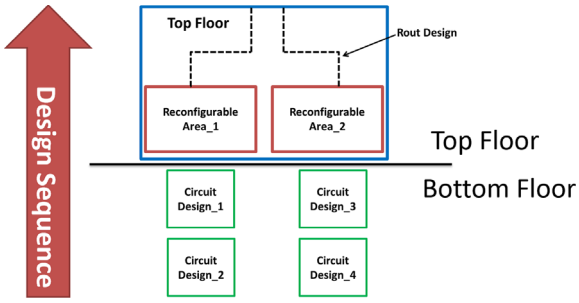


Fig. 6. Bottom-up synthesis.

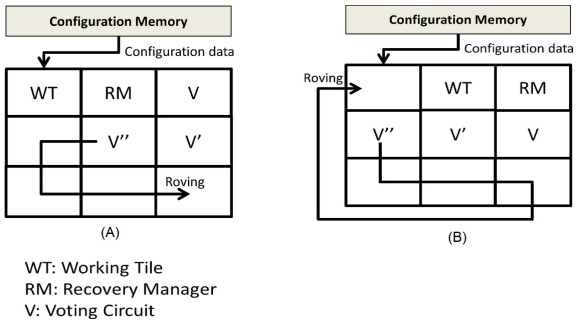


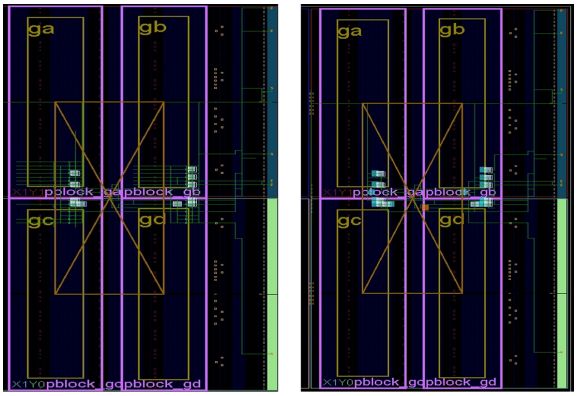
Fig. 7. Proposed Architecture.

3. 提案手法

DPR を用いた回路の耐故障化設計について提案する。そのアーキテクチャの概略図を Fig. 7 に示す。まず，FPGA 内を DPR によりタイル状に分轄し，その中に通常回路（Working Tile）・耐故障動作を制御する回路（Recovery Manager）・故障箇所（Voting Circuit）を検出するための多数決回路を書き込む。この時に使用するコンフィグレーションメモリは，SD カードなどの外部記憶装置とし，FPGA 自身がそこからビットストリームを読み込む。また，コンフィグレーションデータは scrubbing 機能を用いて定期的にリフレッシュされるものとする。タイルに書き込まれた回路デザインは，定期的に隣の領域に上書きされ，それぞれの回路が FPGA 領域内を廻ることで故障を回避する。移動先のタイルは，多数決回路によりあらかじめ正常に動作する事を確認したうえで決定する。本手法は，回路機能を FPGA 領域内で常に巡回させ，信頼性を保証している。特徴として，回路を全て移動させるため，再構成に伴う配線の煩雑化と処理能力の低下を抑えることができる。短所として，それぞれのタイルに実装する，すべての回路デザインをビットストリームとして用意しなければならないため，非常に大容量のコンフィグレーションメモリを必要とする点が挙げられる。また，この手法では 1 つのタイルに 1 つの回路を書き込むため，タイルの粒度が荒くなり，面積効率の点では

Table 1. Implementation Environment.

SoC	XC7Z010-1CLG400C
Evaluation board	Zybo Zynq-7000
Software	Vivado 2015.4
OS	Windows7 Professional(64 bit)
CPU	i7 Q720
RAM	4 GB



(a) Tiles with circuits (b) Black box tiles

Fig. 8. Comparison of writing and removing tiles.

従来のタイルフォールトトレラントのほうが優れているといえる。しかしながら，複数の回路機能を RM 回路 1 つと多数決回路 3 つで耐故障化できるため，耐故障化を施す回路の数が増えるほど面積効率を向上させることができる。

4. DPR を用いたタイルの実装

〈4･1〉 実装回路　まず，本研究で使用した実装環境を Table 1 に示す。本研究では，DPR の提案手法におけるタイルとしての実用性について検証するために，同様の回路デザインを包含した 4 つの DPR 領域をトップフロアに配置した。さらに，Tcl スクリプトを用いてタイル内の回路の巡回を実現するために，タイル内に回路が実装されたビットストリームと，配線がバッファにつながれており機能しない（ブラックボックス化された）ビットストリームをそれぞれ用意し，上書き動作だけで回路デザインの書き込みと削除ができるようにした。それぞれのビットストリームの回路イメージと，これを用いた巡回動作の概略図を Fig. 8 および Fig. 9 に示す。

〈4･2〉 タイルの実装結果　実験の結果，回路は正常に動作し，3 秒おきに回路をタイル間で巡回させることに成功した。しかしながら，DPR によってタイルを実装した結果，タイルサイズの設定において，領域の指定時に課せられる制約により，タイルの粒度を荒くせざるをえないという課題が明らかとなった。

5. 提案手法のスタンドアロン化の検討

本研究で提案している手法は，回路の耐故障動作を自律的に制御できることを最終目標としている。本実装において評価ボードとして使用した Zybo には，ARM コアが FPGA

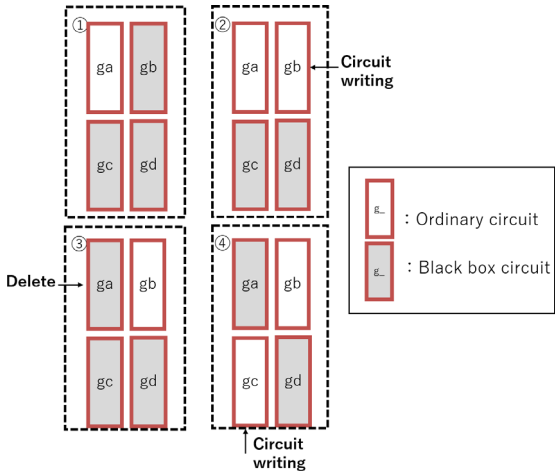


Fig. 9. Cyclic operations.

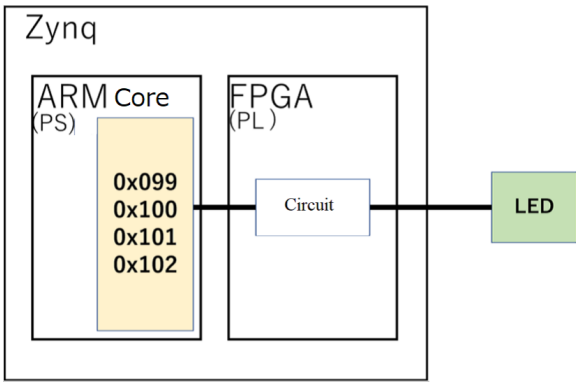


Fig. 10. A configuration of the system using ARM core.

と混載された Zynq-7000 シリーズが搭載されており，プロセッサから FPGA へのアクセスをサポートしている。本研究では，評価ボードに Linux を実装し，このプロセッサを用いた自律制御について検討を行った。ARM コアを用いた自律制御の構成を Fig. 10 に示す。

Zybo に搭載されている Zynq-7000 シリーズは，ARM コアからなる PS 部と FPGA からなる PL 部に分けることができる。PS 部との接続は，PL 部をメモリとしてマッピングすることで実現される。これにより，PS 部からは PL 部はメモリの一部として認識され，通常のメモリと同様の操作で PL 部の回路を使用することができる。

〈5･1〉 システムの設計　検討手法の設計には，ハードウェア部を vivado，ソフトウェア部には Xillinx SDK を用いた。また，PS 部の制御には Linux を用いたため，ARM コア用の Linux カーネルのビルドやデバイスドライバの作成などに Ubuntu を使用した。本実装において作成したファイルとその詳細を Table 2 に示す。これらのファイルは，評価ボードに挿入された microSD カードから読み出される。Zybo は，電源が入ると microSD 中の BOOT.bin をまず発見し，ボード上のメモリにコピーする。そこから読み出される FSBL（First Stage Boot Loader）を使用して，同ファイルに含まれるビットストリームから PL 部の回路を

Table 2. Listing of files.

Files	Contents
BOOT.bin	Boot files as FSBL,Bitstream and u-boot
Devicetree.dth	device tree
Myled.ko	Device driver on PL
uImage	Linux kernel
Uramdisk.image.gz	Disk image

Fig. 11. Operation circuits from Linux.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main()
{
        FILE* fp;
        while(1) {
                fp = fopen("/proc/myled", "w");
                if(fp == NULL) {
                        printf("Cannot open /proc/myled for write¥n");
                        return -1;
                }
                fputs("0x0F¥n", fp);
                fclose(fp);
                sleep(1);
                fp = fopen("/proc/myled", "w");
                if(fp == NULL) {
                        printf("Cannot open /proc/myled for write¥n");
                        return -1;
                }
                fputs("0x00¥n", fp);
                fclose(fp);
                sleep(1);
        }
        return 0;
}
```

Fig. 12. Program code using device driver.

コンパイルし，microSD 中の Linux イメージやデバイスツリーなどを読み込む。PL 部に新たな回路を作成したい場合，BOOT.bin とデバイスドライバを作成し，デバイスツリーに回路情報を加筆することにより実装が可能になる。

〈5･2〉 Linux からの回路操作 Zybo 上で動作する Linux は，Tera term を用いて操作することができる。また，Linux 上からメモリ番地を指定し，データをセットすることで PL 部の回路にデータを与え，操作することができる。Linux 上から PL 部を操作するようすを Fig. 11 に示す。この手法では，Fig. 11 のようにターミナルから直接番地を指定し，データをセットすることも可能であるが，あらかじめデバイスドライバを用意することにより，プログラムを用いた回路動作の自動化が可能になる。本研究では，PL 部に LED を点灯させる回路を実装し，一定間隔で点滅させるプログラムを作成した。ソースコードを Fig. 12 に示す。

本実装により，チップが独立して FPGA 上の回路を操作できる事が確認できた。しかし，PS 部にとって PL 部はメモリの一部としか認識されていないために，本アプローチによる耐故障動作の実現には，さらなる検討が必要であることがわかった。しかし，PL 部への回路書き込みは FSBL によって行われているため，u-boot などの Linux 実装に関連するファイルを排し，ビットストリームと FSBL のみの構成で実装した場合，SD カードから自動的に回路デザインの読み出しが始まり，PL 部に回路を作成することができた。これにより，チップ単体でビットストリームを読み出し，回路を構成することが可能であることが確認できたため，最終目標である FPGA 単体での再構成動作の実現性を示すことができた。

6. ま と め

本研究では，既存の技術を組み合わせて，容易かつ堅牢な耐故障回路を制作するための技術的要素の検討を行った。特に本稿では，提案アーキテクチャの要であるタイルを実装し，Tcl スクリプトによりビットストリームの書き込みと削除の自動化を行い，タイル間を移動する回路動作を再現した。また，Tcl による巡回動作の再現において，ブラックボックス化したビットストリームを用意しておくことで，書き込み動作のみを用いて回路デザインの書き込みと削除が実行できることがわかった。Linux を用いた耐故障動作の自律制御については，評価ボード単体でビットストリームを読み込み，再構成を実行できる事が確認でき，その実現性を示すことができた。

今後は，回路の巡回動作のスタンドアロン化についてさらに検討し，提案手法の実現に必要とされるリカバリマネージャーの検討や scrubbing 技術の検証を行う。さらに，それらを用いた全体的な提案手法の実装を行い，最終的には提案手法の耐故障性の比較及び評価を予定している。

謝 辞

本研究は JSPS 科研費 17K01063 の助成を受けたものである。

文 献

（1） H. Kawai, Y. Yamaguchi, and M. Yasunaga: “Realization of the sound space environment for the radiation-tolerant space craft”, Proc. IEEE Conf. on Reconfigurable Computing and FPGA’s, ReConFig 2006, pp.198–205 (2006)

（2） K. Fujisawa, M. Amagasaki, M. Iida, M. Kuga, and T. Sueyoshi: “A study of run-time fault detection mechanism for fault-tolerant FPGAs”, IEICE Technical Report, Vol.114, No.223, pp.13–18 (2014) (in Japanese)
藤沢賢太郎・尼崎太樹・飯田全広・久我全広・末吉敏則：「フォルトトレラント FPGA 向け実行時故障検出機構の一検討」, 信学会技報, Vol.114, No.223, pp.13–18 (2014)

（3） M. Amagasaki, K. Inoue, Q. Zhao, M. Kuga, and T. Sueyoshi: “Defect-robust FPGA architectures for intellectual property cores in system LSI”, Proc. 23rd Int. Conf. on Field programmable Logic and Applications (FPL2013), pp.1–7 (2013)

（4） K. Inoue, M. Koga, M. Iida, M. Amagasaki, Y. Ichida, M. Saji, J. Iida, and T. Sueyoshi: “An Easily Testable Routing Architecture and Prototype Chip”, *IEICE Trans. on Inf. & Syst.*, Vol.E95-D, No.2, pp.303–313 (2012)

（5） M. Amagasaki, Y. Nishitani, K. Inoue, M. Iida, M. Kuga, and T. Sueyoshi: “A Novel Detection and Recovery Techniques for Physical Defect in FPGA-IP Cores”, *IEICE Trans. on Inf. & Syst.*, Vol.J96-D, No.12, pp.3019–3029 (2013) (in Japanese)

尼崎太樹・西谷祐樹・井上万輝・飯田全広・久我全広・末吉敏則：「システム LSI 搭載 FPGA-IP コア向け物理故障検出および回避方法」，信学論，Vol.J96-D, No.12, pp.3019–3029 (2013)
（6） “Vivado Design Suite User Guide: Partial Reconfiguration”, Xilinx Inc. (2018)
（7） “Vivado Design Suit Tutorial: Partial Reconfiguration”, Xilinx Inc. (2018)
（8） Y. Li, D. Li, and Z. Wang: “A new approach to detect-mitigate-correct radiation-induced faults for SRAM-based FPGAs in aerospace application”, Proc. IEEE 2000 National Aerospace and Electronics Conference (NAECON2000), pp.588–594 (2000)
（9） V. Lakamraju and R. Tessier: “Tolerating operational faults in cluster-based FPGAs”, Proc. ACM/SIGDA eighth international symposium on Field programmable gate arrays (FPGA’00), pp.187–194 (2000)
（10） M. Abramovici, C. Strond, C. Hamilton, S. Wijesuriya, and V. Verma: “Using Roving STARs for On-Line Testing and Diagnosis of FPGAs in Fault-Tolerant Applications”, Proc. IEEE International Test Conference (ITC’99), pp.973–982 (1999)

荻 堂 盛 也（非会員） 2016 年沖縄工業高等専門学校創造システム工学専攻入学。2018 年同高等専門学校創造システム工学専攻修了。同年豊橋技術科学大学大学院工学研究科電気・電子情報工学専攻修士課程入学。現在に至る。

山 田 親 稔（正員） 2000 年琉球大学大学院理工学研究科博士前期課程修了。2004 年同大学大学院博士後期課程単位取得満期修了。同年拓殖大学北海道短期大学専任講師。2007 年沖縄工業高等専門学校情報通信システム工学科助教。2009 年同高等専門学校情報通信システム工学科准教授。2014 年ビクトリア大学（カナダ）客員研究員。2015 年より，沖縄工業高等専門学校情報通信システム工学科准教授。現在に至る。博士（工学）。形式的設計検証，リコンフィギャラブルシステムの研究・教育に従事。IEEE，電子情報通信学会，各会員。

宮 城 桂（非会員） 2008 年高知工科大学情報システム工学科卒業。2010 年同大学大学院修士課程修了。2014 年同大学大学院博士課程修了。同年沖縄工業高等専門学校情報通信システム工学科助教。現在に至る。博士（工学）。自己同期型回路を用いた超低消費電力 VLSI の研究に従事。電子情報通信学会会員。

市 川 周 一（正員） 1985 年東京大学理学部卒業。1987 年同大学大学院理学系研究科修士課程修了。1987 年新技術事業団創造科学推進事業（ERATO）後藤磁束量子情報プロジェクト研究員。1991 年三菱電機（株）LSI 研究所，システム LSI 開発研究所勤務。1994 年名古屋大学工学部助手。1997 年豊橋技術科学大学工学部知識情報工学系講師。2001 年豊橋技術科学大学工学部知識情報工学系助教授。2007 年豊橋技術科学大学工学部知識情報工学系准教授。2010 年豊橋技術科学大学大学院工学系研究科准教授。2011 年沼津工業高等専門学校制御情報工学科教授。2012 年より，豊橋技術科学大学大学院工学系研究科教授。現在に至る。理学博士。並列計算機，並列処理，および専用計算システムアーキテクチャの研究に従事。IEEE（senior member），電子情報通信学会（シニア会員），ACM，情報処理学会，各会員。

藤 枝 直 輝（非会員） 2013 年東京工業大学大学院情報理工学研究科計算工学専攻博士後期課程修了。博士（工学）。同年より豊橋技術科学大学電気・電子情報工学系助教。プロセッサアーキテクチャ，FPGA 応用，組み込みシステム，セキュアプロセッサの研究に従事。情報処理学会，電子情報通信学会，IEEE 各会員。