

A Study of Fault-Tolerant Architecture Using Dynamic Partial Reconfiguration

Seiya Ogido, Chikatoshi Yamada and Kei Miyagi

Dept. of Information and Communication Systems Engineering, National Institute of Technology, Okinawa College
905 Henoko, Nago-shi, Okinawa 905-2192, Japan
Email: {ac164604@edu., cyamada@, k.miyagi@}okinawa-ct.ac.jp

Shuichi Ichikawa

Dept. of Electrical and Electronic Information Engineering, Toyohashi University of Technology
1-1 Hibarigaoka, Tempaku-cho, Toyohashi 441-8580, Japan
Email: ichikawa@tut.jp

Abstract—Processors for embedded products are required to high reliability. Conventionally, it has been remained reliability by the redundant structure of circuits. However, it becomes larger overhead of the circuit area. In this article, we proposed reconfigurable fault tolerant architecture which can recovery from failure status with spare space. Especially, we aim to implement more simple fault tolerant architecture by using structure of dynamic partial reconfiguration. As a result, we discuss implementation of simple fault tolerant architecture.

I. INTRODUCTION

半導体技術の発達により、回路はより微細・高速化が進んでいる。打ち上げてからの修理が不可能な人工衛星や、計器類に高い信頼性が求められる航空機では、こうした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おり、回路の重要な部分に MIL 規格の製品を用いたり、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高い処理能力と引き換えに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面積冗長化回路ほどのスループットは実現できないものの、デバイス上の余白の領域に故障した箇所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、想定する故障の対象として単一縮退故障だけを扱うこととする。プロセッサの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指して、回路の動作を止めること無く信頼性を保証することができる DPR(動的部分再構成)と、Scrubbingなど複数の既存技術を組み合わせた再構成型耐故障アーキテクチャの検討を行った。

以下では、先行研究 [8][9] で提案されているフォルトトレラント 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(動

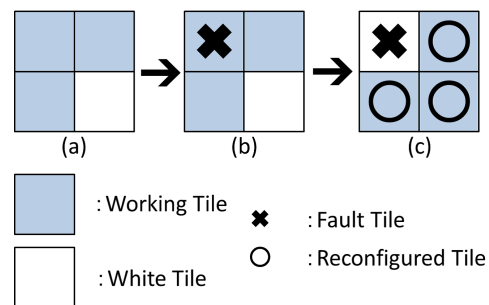


Fig. 1. Tile fault tolerant

的部分再構成)[10][11] 回路の実デバイス上へ行った実装について説明する。

II. RELATED WORKS

A. Tile fault tolerant

再構成型耐故障アーキテクチャの代表的な例として、タイルフォルトトレラント [3][4] について説明する。本手法は、デバイス内のプログラマブル領域が LB(Logic Block) と配線が格子状に配置されたアイランドスタイル構造の FPGA において、格子の一つ一つまたは複数のセルをまとめたものを、タイルと呼ばれるユニットとみなすものである。すなわち、FPGA 内をタイルによって分割した構造とし、デバイス上に作成される回路は、タイルの集合体として表現される。タイルフォルトトレラントの特徴として、再構成動作を回路の書き込まれていない余白部分のタイルを用いて行うため、外部から追加のリソースを必要とせず、面積効率に優れている点が挙げられる。問題点として、再構成の度にタイル同士をつなぐ配線のパターンが複雑になっていき、配線遅延による処理能力が低下していく事が挙げられる。タイルフォルトトレラントの復帰動作の概略図を Fig.1 に示す。

B. Roving STAR

次に、タイルを用いた回路の検査手法について説明する。代表的なものとして、先行研究 [5] で提案されている Roving

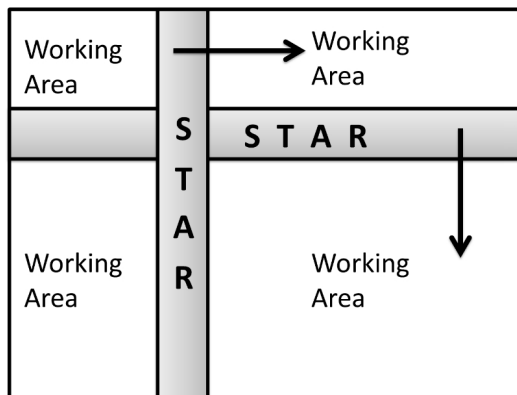


Fig. 2. Roving STAR

STAR が挙げられる。STAR とは Self-Testing Area の略であり、通常の実行中の領域とは独立した検査中の領域のことを指す。

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 により生成されたの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ないで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。

III. PREVIOUS WORKS

A. FT-FPGA

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 [6][7][8] で提案されている FT-FPGA 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 アーキテクチャの概略図を Fig.3 に示す。

FT-FPGA 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 Block), CB (Connection Block) に分けることができる。LB には LUT (Look Up Table) や BLE (Basic Logic Element) などが格納されており、回路を構成するための基本的な論理が実装されている。LB 同士の接続は SB と CB により行われる。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 (Tail 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 $n \times n$ の場合は、 TA_n 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

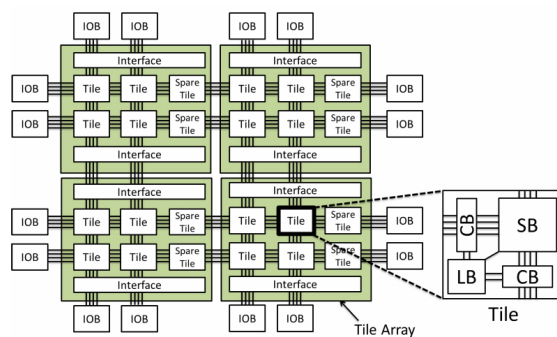


Fig. 3. Structure of FT-FPGA

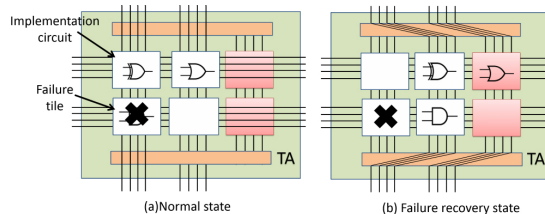


Fig. 4. Reconfiguration operations

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。

次に、FT-FPGA の故障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ト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。FT-FPGA の故障回復・検出の動作を Fig.4 および 5 に示す。

次に FT-FPGA による故障検出動作について説明する。Fig.5 は、アレイサイズ TA_2 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 1 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 [6][7][8] 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。

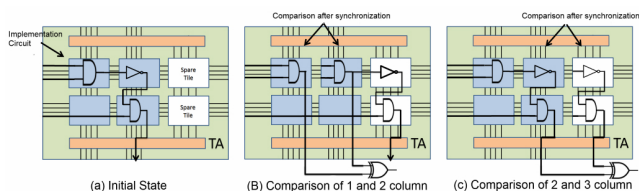


Fig. 5. TA operation

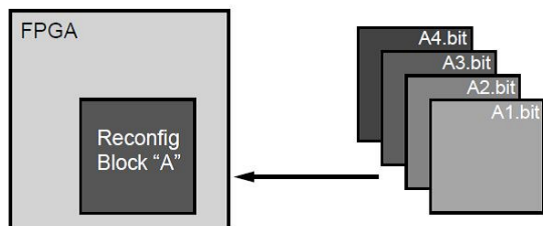


Fig. 6. Architecture of DPR

B. 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[10][11] 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。DPR のアーキテクチャを Fig.6 に示す。

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、結線情報等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.7 に示す。

IV. PROPOSED METHOD

次に、既存手法を組み合わせた回路の耐故障手段について提案する。アーキテクチャの概略図を Fig.8 に示す。

本提案手法では、FPGA 内を DPR によりタイル上に分轄し、その中に耐故障化を施す回路・耐故障動作を制御する回路・故障箇所を検出するための多数決回路が書き込まれた構造になっている。この時に際使用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、そこから FPGA 自身が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

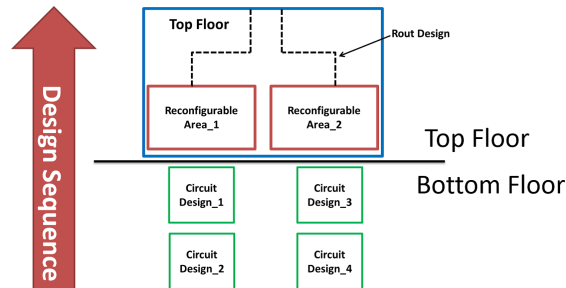


Fig. 7. Bottom-up synthesis

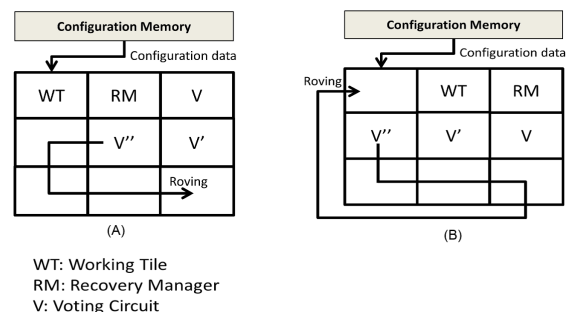


Fig. 8. Architecture of the proposed method

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かを確認したうえで決定する。

この手法は、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。

例を挙げると、FPGA 領域内をパーティションによりタテとヨコに 4 分轄した場合は最低でも 4 つ、それぞれの領域に 2 種類の回路を構成する場合は 8 つのビットストリームを用意しておく必要がある。

V. IMPLEMENTATION OF DPR

A. circuit scale

まず、今回の実装に使用した実装環境を Table I に示す。本研究では、通常回路の DPR による耐故障化が容易であるかを検証するために、予め上階層 (スタティック領域) に

TABLE I
IMPLEMENTATION ENVIRONMENT

SoC	XC7Z010-1CLG400C
評価ボード	Zybo Zynq-7000
ソフトウェア	Vivado 2015.4
OS	Windows7 Professional(64bit)
CPU	i7 Q720
RAM	4GB

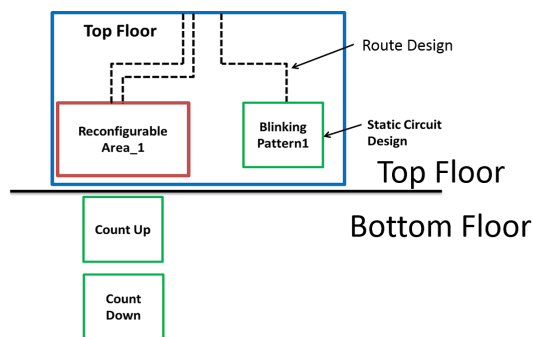


Fig. 9. Initial state

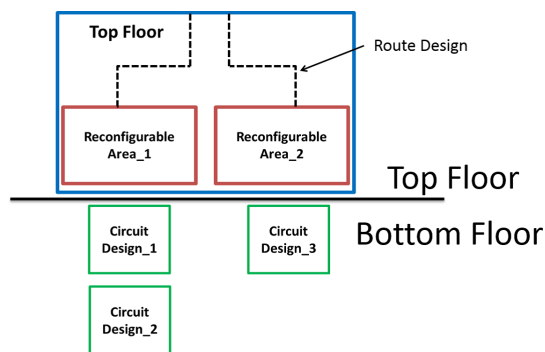


Fig. 10. Creating a reconfigurable circuit

TABLE II
EXPERIMENTAL RESULT

	Top floor in initial state	Top floor in DPR state	Reconfigurable circuit (Area "gb")
FF	28	-	28
LUT	38	1	37
IO	7	7	-
CLK	1	-	-

回路が書き込まれたものを用意した。リコンフィギュラブル化を施す前の回路の概略図を Fig. 9 に示す。

初期状態では、比較のために上階層に DPR 領域と通常回路をそれぞれ設けた。さらに、最上位階層から通常回路を取り除いた後に、DPR 領域を新たに設け、そこに取り除いた回路が書き込まれるように設計しなおした。リコンフィギュラブル化を施した回路の概略図を Fig.10 に示す。

B. Experimental result

実デバイス上で通常回路をリコンフィギュラブル化したようすを Fig.11 および Fig.12 に示す。

初期状態では、プログラマブル領域の右下に通常回路(スタティック回路)が実装されているが、Fig.12 では、所望の DPR 領域が追加され、そこに回路が実装されていることがわかる。本実装における回路規模の変化を TableII に示す。

Table II において、左端の列が初期状態での回路規模を示しており、その右隣が通常回路を削除した場合のスタティック領域の回路規模、右端は、取り除いた回路を再び付加した時の DPR 領域内の回路規模である。リコンフィギュラブル化によって、クロックが削除されてしまっているように

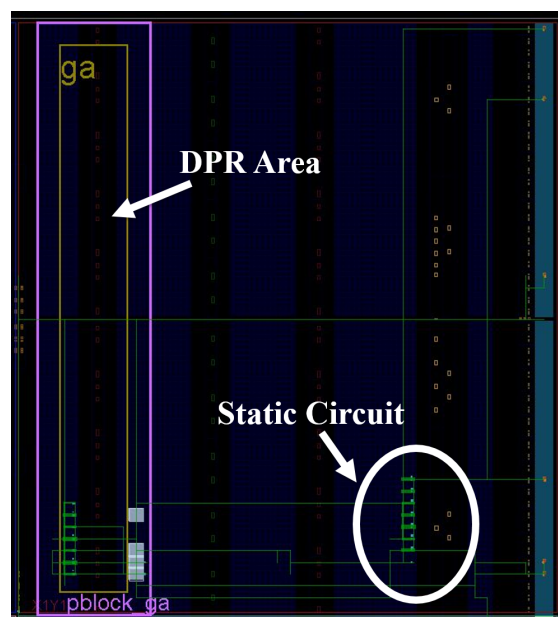


Fig. 11. Initial state

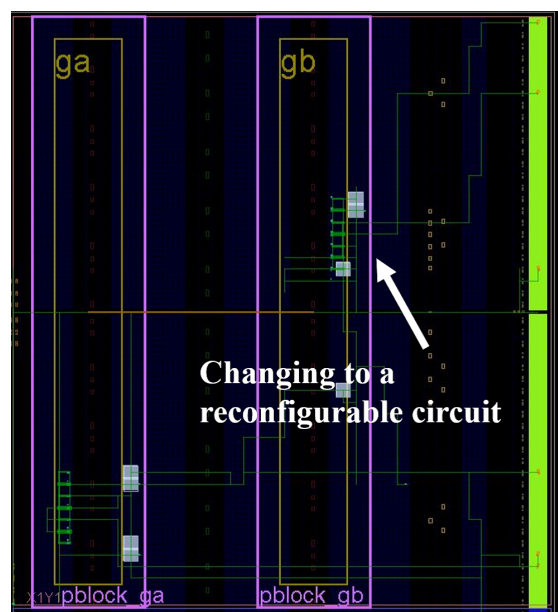


Fig. 12. Changing to a reconfigurable circuit

見えるが、比較用に用意した DPR 領域内(領域 ga)にクロックが含まれており、この表からは見えなくなってしまっている。Table II から、リコンフィギュラブル化を施した回路の規模が初期状態の回路規模と等しいことがわかり、リコンフィギュラブル化における面積オーバーヘッドが無いことが確認できた。

通常回路のリコンフィギュラブル化については、FPGA 上に実装される各回路をあらかじめ徹底的にモジュール化しておくことで、それらの回路をボトムアップ型合成における下層回路と同様に扱うことができることが確認できた。

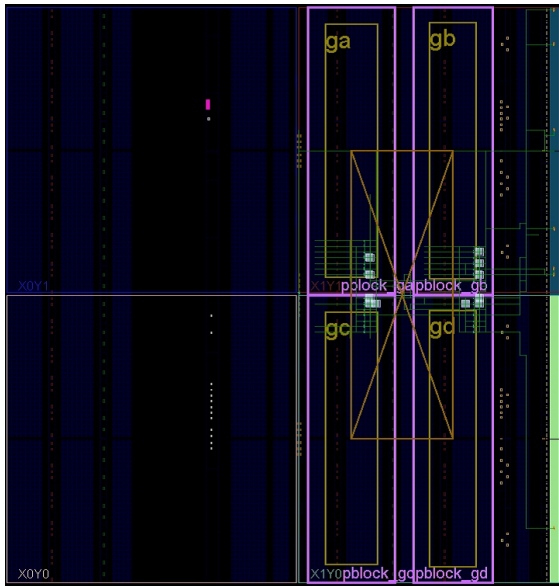


Fig. 13. Tiled DPR

C. Create tile by DPR

さらに、DPR が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。

Fig.13 からわかるように、トップフロアは大きく4つの領域に分割されたデザインになっている。本研究では、右半分領域を利用して4つのDPR領域を右半分にまとめて配置し、タイルとした。DPRの場合、トップフロアの上下端を領域内に収めれば、領域のサイズを自由に設定することができる。面積効率を考えると、1つ当たりのタイルの大きさはできるだけ小さいほうが有利といえる。しかし、ブロックRAMなど、FPGA領域には特定の箇所まとめて配置されている部品が存在するため、同様の回路を各タイルで実装する場合、タイル1つあたりの領域サイズを安易に小さくすることはできない。提案手法を実現するには、タイルに実装する回路の規模・動作を踏まえた上でDPR領域の大きさを設定する必要がある。

VI. CONCLUSION

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、通常回路のリコンフィギュラブル化と提案手法におけるDPRの実用性について検証するために、DPRの実デバイス上への実装を行なった。その結果、耐故障化が施されていない回路を想定してスタティック領域に実装された回路を、DPR領域内に移し、実装することができた。その際、DPRによるリコンフィギュラブル化において、面積オーバーヘッドが無いことが確認できた。DPR領域をタイル状に配置し、提案手法への有用性を確認することができた。今後は、DPRを用いたタイルの実装を行い、提案手法の実現に必要なscrubbingなどの既存技術の検証と、それらを用いた提案手法の実装を行い、最終的には提案手法の耐故障性能の評価を行っていく予定である。

REFERENCES

- [1] H. Kawai, Y. Yamaguchi, and M. Yasunaga, "Realization of the sound space craft", in ReConFig'06, Sept. 2006, pp. 198-205.
- [2] Andr DeHon and Nikil Mehta, "Exploiting Partially Defective LUTs : Why You Don't Need Perfect Fabrication", Proc. FPT, December 2013.
- [3] Y. Li, D. Li and Z. Wang, "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] V. Lakamraju and R. Tessier, "Tolerating operational faults in cluster-based fpgas", Proc. FPGA, February 2000.
- [5] 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", Test Conference, 1999. Proceedings. International, pp. 973-982, (1999).
- [6] M. Amagasaki, K. Inoue, Q. Zhao, M. Kuga and T. Sueyoshi, "DEFECT-ROBUST FPGA ARCHITECTURES FOR INTELLECTUAL PROPERTY CORES IN SYSTEM LSI", Proc. FPL, September 2013.
- [7] 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, February 2012.
- [8] FUJISAWA Kentaro, AMAGASAKI Motoki, IIDA Masahiro, KUGA Morihito and SUEYOSHI Toshinori: "A study of run-time fault detection mechanism for fault-tolerant FPGAs," IEICE Technical Reports, vol.114(223), pp.13-18 (2014).
- [9] AMAGASAKI Motoki, NISHITANI Yuki, INOUE Kazuki, IIDA Masahiro, KUGA Morihito and SUEYOSHI Toshinori: "A Novel Detection and Recovery Techniques for Physical Defect in FPGA-IP Cores," IEICE Trans. vol.J96-D, no.12, pp.3019-3029 (2013).
- [10] "Vivado Design Suite Tutorial Partial Reconfiguration," Xilinx Inc. (2015).
- [11] "Vivado Design Suit Tutorial Partial Reconfiguration", Xilinx 社, 2015, Novemver 18, pp. 5-26.