HAL open science

Atténuation des Défauts dans les Réseaux sur Puce avec une Approche de Brassage de Bits Basée sur des Régions

Romain Mercier, Cédric Killian, Angeliki Kritikakou, Youri Helen, Daniel Chillet

To cite this version:

Romain Mercier, Cédric Killian, Angeliki Kritikakou, Youri Helen, Daniel Chillet. Atténuation des Défauts dans les Réseaux sur Puce avec une Approche de Brassage de Bits Basée sur des Régions. GRETSI 2022 - XXVIIIème Colloque Francophone de Traitement du Signal et des Images, Sep 2022, Nancy, France. pp.1-4. hal-03926136

HAL Id: hal-03926136

https://inria.hal.science/hal-03926136v1

Submitted on 6 Jan 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Atténuation des Défauts dans les Réseaux sur Puce avec une Approche de Brassage de Bits Basée sur des Régions.

Romain MERCIER[1], Cédric KILLIAN[1], Angeliki KRITIKAKOU[1], Youri HELEN[2], Daniel CHILLET[1]

[1]Univ Rennes, Inria, CNRS, IRISA
263 Av. Général Leclerc, 35000 Rennes, France
prenom.nom@irisa.fr

[2]DGA MI
136 La Roche Marguerite, 35170 Bruz, France
youri.helen@intradef.gouv.fr

Résumé – L'évolution technologique s'appuyant sur la réduction de la taille des transistors conduit à une plus grande sensibilité pouvant conduire à des fautes. De plus, au sein d'un système sur puce, un réseaux d'interconnexion fautif peut largement impacter le fonctionnement d'une application puisqu'il transporte une grande quantité de données. Compte tenu des ces deux éléments, des techniques de protection des données doivent être développées pour atténuer l'impact des fautes sur l'application. Des travaux proposent de mélanger les bits à l'intérieur d'un flit, transférant l'impact des fautes sur les bits les moins significatifs. Cependant, ces approches sont appliquées à un niveau de grain fin, fournissant une atténuation des défauts efficace, mais avec des coûts matériels importants. Pour remédier à cette limitation, ce travail propose une technique de brassage de bit par région, qui sacrifie l'efficacité afin de réduire les coûts matériels.

Abstract – The technological evolution based on the reduction of the transistor size leads to a greater sensitivity leading to faults. Moreover, in a system-on-chip, a faulty interconnect network can largely impact the operation of an application since it carries a large amount of data. Given these two elements, data protection techniques must be developed to mitigate fault impacts on the application. Some works propose to shuffle the bits inside a flit, transferring the fault impacts on the least significant bits. However, these approaches are applied at a fine-grained level, providing effective fault mitigation, but with significant hardware costs. To address this limitation, this work proposes a region-based bit-shuffling technique which sacrifices efficiency in order to reduce hardware costs.

1 Introduction

Dans l'ère technologique actuelle, les NoCs sont devenus plus sensibles aux défauts permanents, qui sont dûs aux processus de fabrication [1], aux défauts de vieillissement [2] et aux radiations [3]. Alors que les défauts de fabrication et de vieillissement sont des sources bien connues de perturbations de bit unique (*Single Bit Upsets* - SBUs), c'est-à-dire plusieurs bits affectés par plusieurs évènements uniques (*Single Event Upsets* - SEUs), les radiations peuvent conduire à des SBUs et des perturbations de bit multiple [4], c'est-à-dire plusieurs bits affectés par un seul SEU.

Dans ce contexte, des techniques de tolérance aux fautes sont couramment appliquées sur les NoCs pour corriger les fautes permanentes [1, 5, 6]. Cependant, ces méthodes induisent souvent un surcoût matériel important, en termes de surface et de consommation d'énergie, et peuvent impacter les performances du NoC de manière drastique jusqu'à un déni de communication lorsque des IPs deviennent inaccessibles. Le calcul approximatif peut également être utilisé pour atténuer les défauts dans les NoCs. Dans [7], une technique de brassage de bits, nommée BiSu, exploite la position des défauts permanents au moment de l'exécution et change l'ordre des bits à l'intérieur d'un flit dans le but de réduire l'impact des défauts. Cependant, cette approche nécessite un grand nombre de blocs matériels supplémentaires dans les NoCs.

Pour réduire les coûts matériels, tout en assurant la protection des données, nous proposons une technique de brassage des bits basée sur des régions (R-BiSu). Pour cela, le NoC est divisé en régions, et la technique de brassage de bits est ensuite appliquée à leurs frontières. Nous proposons une méthode hiérarchique pour calculer le brassage de bits de la région $X \times X$ à partir du brassage de bits de la région $(X-1) \times (X-1)$. Nous concevons un bloc matériel qui calcule les valeurs des registres de brassage, au lieu d'utiliser le noyau du bloc dédié des routeurs. Enfin, nous étudions en détail le compromis entre le niveau de protection fourni et les coûts matériels requis par l'approche R-BiSu.

2 Approche Proposée

Cette section présente l'approche du brassage de bits par région (R-BiSu) qui vise à réduire les coûts matériels. La section 2.1 présente le domaine cible et les hypothèses. La section 2.2 fournit les bases de la technique BiSu, tandis que les

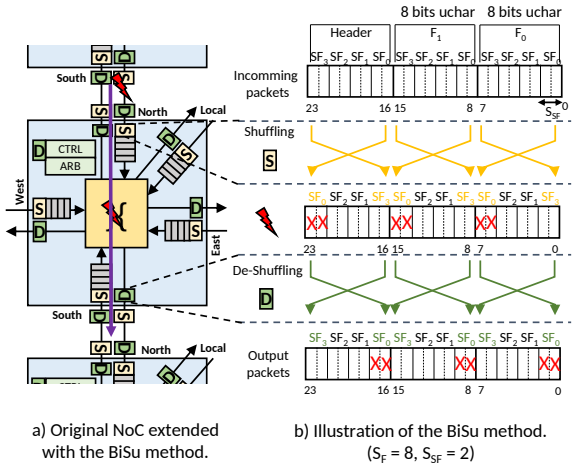


FIGURE 1 – NoC étendu avec la méthode BiSu.

détails sont disponibles dans [7]. Enfin, la méthode R-BiSu est décrite dans la section 2.3.

2.1 Domaines cibles et hypothèses

La méthode BiSu atténue les défauts permanents multiples qui peuvent survenir dans les chemins de données du NoC. Comme l'illustrent les éclairs rouges de la Figure 1, les défauts peuvent être localisés dans i) les interconnexions entre routeurs, ou dans ii) les mémoires tampons et l'aiguillage présents dans chacun des routeurs. Dans ce travail, nous supposons que les positions des défauts (décrites par un masque d'erreurs) sont fournies par des méthodes telles que les techniques d'autotest intégré [8]. Étant donné que l'objectif de l'approche proposée est de réduire l'impact des fautes multiples, au lieu de les corriger, les domaines ciblés sont les applications résilientes aux erreurs, utilisées dans les domaines de l'informatique et des communications approximatives [9].

2.2 Contexte : La technique BiSu Basique

La technique de base Bit-Shuffling (BiSu) est mise en œuvre en ajoutant au NoC des blocs matériels supplémentaires, c'est-à-dire des blocs de brassage (*Shuffler* - S) et de débrassage (*Deshuffler* - D), comme le montre la Figure 1a. Les flits de taille S_F bits, qui transitent sur le NoC, sont divisés en N_{SF} blocs, appelés sous-flits (*SubFlits* - SFs). Chaque SF est constitué de S_{SF} bits. Le bloc de brassage réorganise les SFs dans le but de minimiser l'impact des fautes. Le bloc de débrassage ramène l'ordre initial des SFs, comme illustré sur la Figure 1b. La technique BiSu est appliquée pour atténuer les erreurs sur les bus d'interconnexion et à l'intérieur des routeurs. Un bloc D supplémentaire est ajouté dans le contrôleur de routage (CTRL) pour lire les informations de routage de l'en-tête brassé et transférer le paquet courant vers la sortie attendue. De plus amples détails sont donnés dans [7], notamment concernant la protection des flits critiques, comme par exemple les flits d'en-tête.

2.3 Approche du Brassage de Bits par Région

Dans ce travail, nous proposons une technique de brassage de bits basée sur des régions afin de réduire les coûts surfacique et de consommation d'énergie en relâchant l'efficacité de la technique BiSu. L'idée de base de R-BiSu est de diviser le NoC en régions de routeurs, qui sont globalement protégées par la méthode BiSu.

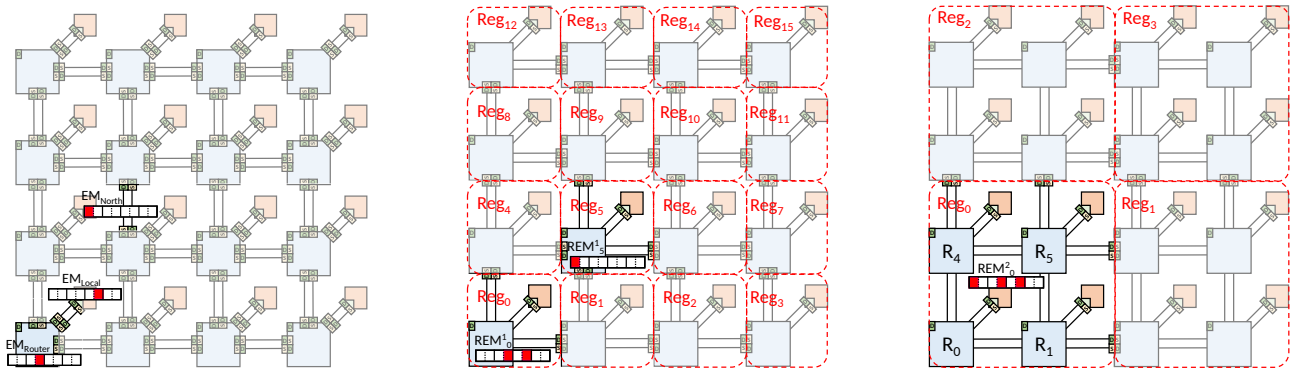
La Figure 2 illustre la notion de régions pour un NoC de taille 4×4 routeurs, où les régions sont distinguées par les carrés en pointillés rouges. La technique BiSu basique, c'est-à-dire la taille 0, protège chaque interconnexion et chaque routeur comme le montre la Figure 2a. Les Figures 2b et 2c représentent respectivement une configuration R-BiSu de taille 1 et 2, c'est-à-dire que le NoC est respectivement divisé en régions de tailles 1×1 et 2×2 routeurs, incluant les IPs locaux. En comparant les techniques BiSu et R-BiSu, on peut observer que la technique BiSu nécessite 304 blocs S et D, alors que les configurations 1×1 et 2×2 réduisent respectivement le nombre de blocs S et D à 164 et 80, réduisant ainsi la surface et la consommation d'énergie.

2.3.1 Calcul des Masques d'Erreur des Régions

La méthode R-BiSu utilise les informations relatives à l'état de la région, données par le Masque d'Erreur de la Région (*Region Error Mask* - REM), afin de réduire autant que possible l'impact des défauts. Pour cela, on utilise une méthode hiérarchique qui calcule le masque d'erreur de la région $X \times X$ sur la base des masques d'erreur des régions $(X-1) \times (X-1)$. Une fois le REM final obtenu, les registres des blocs S et D peuvent être calculés avec un faible coût matériel, comme le montre la section 3. La Figure 2 montre en exemple un NoC de taille 4×4 affecté par trois fautes en considérant des flits de 8 bits et des SFs de 2 bits. Les masques d'erreurs associés sont affichés et indiquent les fautes (mises en évidence par une couleur rouge) pour les différentes tailles de régions. Plus d'informations concernant le calcul de ces masques d'erreurs peuvent être trouvées dans [10].

2.3.2 Calcul des Registres de Brassage et de Débrassage

Le calcul des registres des blocs S et D de la méthode BiSu basique est effectué par le noyau dédié des IPs. Les résultats sont distribués aux blocs S et D par des flits dédiés. Bien que le calcul par le cœur dédié des IPs permette de réduire les coûts matériels, il augmente la charge de travail sur ces derniers puisque le calcul des registres s'ajoute à la charge de travail standard de l'application. Afin de remédier à ce problème, nous concevons un bloc matériel dédié à l'intérieur des régions qui calcule la valeur des registres dont le rôle est de contrôler les multiplexeurs des blocs S et D.



(a) Taille 0 (b) Taille 1×1 (c) Taille 2×2

FIGURE 2 – Illustration de la technique R-BiSu pour différentes tailles de régions.

3 Résultats Expérimentaux

Cette section présente les résultats en termes d'efficacité et de coûts matériels. Tout d'abord, nous présentons le dispositif expérimental. Ensuite, nous analysons l'impact de la taille des régions sur l'efficacité de l'approche R-BiSu, lorsque les données voyagent sur un NoC défectueux, en utilisant la métrique MSE. Enfin, nous présentons les coûts matériels en termes de surface et de consommation électrique. R-BiSu a un impact plus faible que BiSu sur la latence, qui est négligeable, comme le montre [7].

3.1 Configuration expérimentale

Pour nos expériences, nous considérons un NoC 8×8 utilisant l'algorithme de routage XY, où la méthode R-BiSu est implémentée en considérant des régions carrées. Des paquets de 16 flits sont injectés selon le modèle d'injection TORNADO, où chaque IP envoie un paquet à chaque autre IP. Les défauts sont injectés de manière aléatoire dans le chemin de données du NoC en utilisant le modèle de faute stuck-at [11]. De plus, nous considérons que les fautes injectées ont toujours un impact sur les données en appliquant une inversion de bit. De cette façon, l'effet de masquage dû aux valeurs des données est évité et l'impact des fautes est toujours visible. La métrique MSE est utilisée pour quantifier l'efficacité de la méthode R-BiSu sur la base de 10000 ensembles d'injection de fautes. Le routeur CONNECT [12] est utilisé comme référence pour l'implémentation matérielle. Les résultats sont synthétisés sur une technologie FDSOI de 28 nm grâce aux outils de synthèse de haut niveau de Mentor Graphic, pour une fréquence d'horloge de 1 GHz.

3.2 Résultats sur l'efficacité

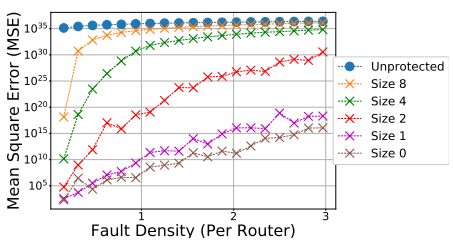
La Figure 3a représente la MSE en fonction de la densité de défauts pour différentes tailles de régions en considérant des flits de 64 bits divisés en SFs de 4 bits. Sur cette figure, on peut observer que la taille des régions a un impact sur l'efficacité de la méthode R-BiSu. On observe que la MSE augmente avec la taille de la région, jusqu'à atteindre les mêmes résultats qu'avec un NoC non protégé, lorsque la densité de défauts est élevée. Cependant, nous observons que la taille des régions peut être augmentée de 0 à 1 avec un faible impact sur la MSE.

3.3 Résultats sur le coût matériel

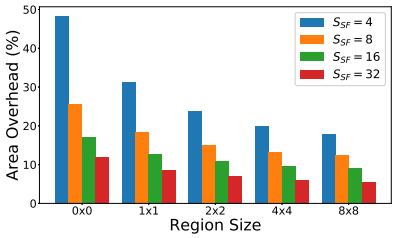
Les Figures 3b et 3c présentent respectivement les surcoûts surfacique et de puissance de la méthode R-BiSu pour différentes tailles de région et de SF. On peut également observer que le bloc matériel pour le calcul des registres de brassage et de débrassage a un faible impact sur le coût matériel global de la méthode. Par exemple, les surcoûts surfacique et de puissance sont respectivement augmentés de $5,3\%$ et $5,9\%$ pour des régions de taille 1. La latence pour calculer les registres dépend du nombre de SFs. Par exemple, si le flit est composé de 16 SFs, alors la latence pour mettre à jour les registres est égale à 370 ns et respectivement de 120 ns et 44 ns pour 8 et 4 SFs. Par conséquent, ces blocs matériels ont un faible coût matériel et peuvent être utilisés pour relâcher la pression de calcul sur les cœurs dédiés des IPs. De plus, ces blocs peuvent être partagés entre plusieurs régions afin de réduire davantage les coûts matériels.

3.4 Arbitrage entre coûts et efficacité

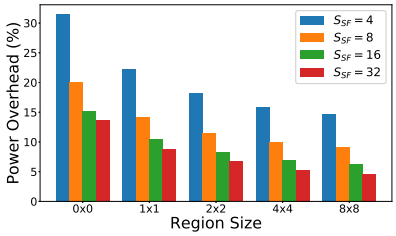
Les Figures 3d et 3e tracent respectivement le front de Pareto surfacique et de puissance pour différentes tailles de SFs et de région, en considérant des flits de 64 bits et une densité de défauts égale à 1.00 défaut par routeur. C'est-à-dire que l'on injecte aléatoirement un nombre de faute égale au nombre de routeur qui compose le NoC. D'après les résultats obtenus, nous observons qu'il existe des cas où BiSu n'appartient pas au front de Pareto. De plus, on observe que la taille des régions peut être augmentée de 0 à 2 avec un faible impact sur l'efficacité, réduisant respectivement les surcoûts surfacique et de puissance de 48% à 33% et de 34% à 22%. Enfin, nous pouvons conclure que l'augmentation de la taille des régions réduit davantage le surcoût matériel que l'augmentation de la taille des SFs.



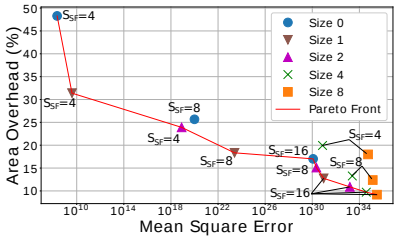
(a) Étude de l'erreur quadratique moyenne



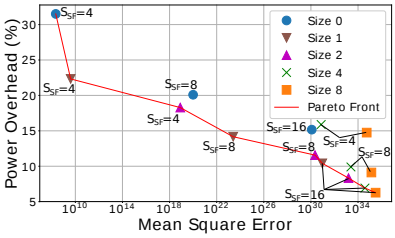
(b) Comparaison de la surcharge de surface



(c) Comparaison de la consommation d'énergie



(d) Mise en évidence du front de Pareto de la zone



(e) Mise en évidence du front de Pareto de la puissance

FIGURE 3 – Résultats expérimentaux de la méthode R-BiSu à l'échelle du NoC. ($S_F = 64, S_{SF} = 4$)

4 Conclusion

Dans ce travail, nous avons proposé une méthode de brassage de bits basée sur des régions dans le but d'atténuer les fautes avec une réduction du coût matériel par rapport à la méthode BiSu de base. Pour y parvenir, une méthode hiérarchique est proposée afin de calculer les masques d'erreur d'une région complète. De plus, nous avons conçu un bloc matériel pour calculer les registres des blocs de brassage et de débrassage, dans le but de relâcher la pression sur les cœurs dédiés des IPs. Les résultats obtenus montrent qu'une augmentation de la taille des régions de 0 à 2 peut réduire les coûts matériels avec un faible impact sur l'efficacité.

Remerciements

Ce travail est soutenu par la Direction Générale de l'Armement (DGA) et le projet ANR SHNoC, subvention ANR-18-CE25-0006 de l'Agence Nationale de la Recherche française.

Références

[1] E. Dubrova, *Fault-Tolerant Design*. Fault-Tolerant Design, Springer, 2013.

[2] S. Kundu and S. Chattopadhyay, *Network-on-Chip : The Next Generation of System-on-Chip Integration*. Taylor & Francis, 2014.

[3] "Space Product Assurance : Techniques for Radiation Effects Mitigation in AASIC and FPGAs Handbook," tech. rep., ESA Requirements and Standards Division, Sept. 2016.

[4] Mutuel, "Single Event Effects Mitigation Techniques Report," *Federal Aviation Admin., William J. Hughes Tech. Center*, Feb. 2016.

[5] Z. Chen, Y. Zhang, Z. Peng, and J. Jiang, "A Deterministic-Path Routing Algorithm for Tolerating Many Faults on Wafer-Level NoC," in *Des. Automat. Test in Europe Conf. Exhib. (DATE)*, pp. 1337–1342, Mar. 2019.

[6] K. Khalil, O. Eldash, A. Kumar, and M. Bayoumi, "Self-Healing Hardware Systems : A Review," *Microelectron. J.*, vol. 93, p. 104620, 2019.

[7] R. Mercier, C. Killian, A. Kritikakou, Y. Helen, and D. Chillet, "BiSuT : A NoC-Based Bit-Shuffling Technique for Multiple Permanent Faults Mitigation," *IEEE Trans. on Comput.-Aided Des. of Integr. Circuits and Syst. (TCAD)*, pp. 1–1, 2021.

[8] H. J. Mohammed, W. N. Flayyih, and F. Z. Rokhani, "Tolerating Permanent Faults in the Input Port of the Network on Chip Router," *J. of Low Power Electron. and Appl.*, vol. 9, pp. 1–11, Feb. 2019.

[9] A. B. Ahmed, D. Fujiki, H. Matsutani, M. Koibuchi, and H. Amano, "AxNoC : Low-power Approximate Network-on-chips Using Critical-path Isolation," in *IEEE/ACM Int. Symp. on Networks-on-Chip (NOCS)*, no. 6, pp. 1–8, Oct. 2018.

[10] R. Mercier, C. Killian, A. Kritikakou, Y. Helen, and D. Chillet, "A Region-Based Bit-Shuffling Approach Trading Hardware Cost and Fault Mitigation Efficiency," in *Int. Symp. on Defect and Fault Tolerance in VLSI and Nanotechnol. Syst. (DFT)*, pp. 205–212, IEEE, Oct. 2021.

[11] W. J. Dally and B. P. Towles, *Principles and Practices of Interconnection Networks*. Elsevier, 2004.

[12] M. K. Papamichael and J. C. Hoe, "CONNECT : Re-examining Conventional Wisdom for Designing Nocs in the Context of FPGAs," in *ACM/SIGDA Int. Symp. Field Program. Gate Arrays (FPGA)*, pp. 37–46, Feb. 2012.