



HAL
open science

High-Level Reliability Evaluation of Reconfiguration-Based Fault Tolerance Techniques

Tien Thanh Nguyen, Mathieu Thevenin, Anthony Mouraud, Gwenolé Corre,
Olivier Pasquier, Sébastien Pillement

► **To cite this version:**

Tien Thanh Nguyen, Mathieu Thevenin, Anthony Mouraud, Gwenolé Corre, Olivier Pasquier, et al.. High-Level Reliability Evaluation of Reconfiguration-Based Fault Tolerance Techniques. Reconfigurable Architectures Workshop (RAW 2018), May 2018, Vancouver, Canada. pp.202-205, 10.1109/IPDPSW.2018.00038 . hal-01797245

HAL Id: hal-01797245

<https://hal.science/hal-01797245v1>

Submitted on 10 Feb 2025

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.

High-Level Reliability Evaluation of Reconfiguration-Based Fault Tolerance Techniques

Tien Thanh Nguyen^{*‡}, Mathieu Thevenin[†], Anthony Mouraud^{*}, Gwenole Corre[†],
Olivier Pasquier[‡] and Sebastien Pillement[‡]

^{*} CEA/CTREG/DPLOIRE, 5 rue de l'Halbrane, 44340 Bouguenais, France; Email: firstname.lastname@cea.fr

[†] CEA Saclay, 91191 Gif-Sur-Yvette, Essonne Cedex, France; Email: firstname.lastname@cea.fr

[‡] IETR, Universit de Nantes, 44306 Nantes Cedex, France; Email: firstname.lastname@univ-nantes.fr

Abstract—Field Programmable Gate Arrays (FPGAs) offer very efficient fault-tolerance strategies based on partial reconfiguration. These strategies use dedicated spare resources at different levels to replace faulty elements. A method to determine if the targeted level of reliability can be obtained is then required. In this paper, we generalize the reliability estimation formulas for the state-of-the-art fault tolerance techniques based on partial reconfiguration. We use those formulas to estimate the bounds of the different techniques in terms of reliability level and bitstream overhead. This work makes possible to evaluate early in the design-flow the achievable mission time for a given number of faults per time unit. Thanks to its generality, our approach can be derived for future new partial reconfiguration approaches.

I. INTRODUCTION

Nowadays, the FPGAs are widely used in aircraft, military devices or automotive. Most of these circuits use an SRAM configuration memory which is extremely sensitive to energy deposits (cosmic rays, neutrons etc.), to electromigration or other causes of defects. We intend in this work to evaluate the mitigation techniques deployed using the Field Programmable Gate Array (FPGA) partial reconfiguration ability of modern chips. The detection and localization of faults is already studied in the literature [1], [2] and is beyond the scope of this work.

The default model used, defines transient faults which can be corrected [3] and permanent faults, due to partial destruction, that remain active once they appeared. The avoidance of the permanent faulty elements can be performed by partial reconfiguration of the FPGA [4]. It is achieved by synthesizing several alternative configurations of the modules in order to map them on spare resources. Storing all these alternatives modules may require a lot of memory resources, but the number of alternatives impacts also the reliability level.

Reliability level estimation for FPGAs has been already investigated in previous works [5]. The authors assumed that any faulty element can be replaced by any alternative spare resource and they considered modules of identical sizes (homogeneous approach). We then consider this case as the best-case scenario. Indeed, a spare resource cannot always be used to replace a faulty element due to localization or interfaces problems. Moreover, an application comprises heterogeneous requirements implemented into modules of different size. Consequently, a gap exists between the best-case scenario

evaluated by the previous works and the real-conditions. Furthermore, bitstream size overhead cannot be neglected since the probability of a fault occurring in a memory is directly correlated to its area. This evaluation is not supported in previous works to the best of our knowledge.

Our contributions can be summarized as 1) the generalization of the formulas for the evaluation of the reliability level to the heterogeneous designs case; 2) the evaluation of the bitstream size overhead; 3) a classification of the various partial reconfiguration approaches and 4) the results of our estimations on examples drawn from the literature applied to a Virtex-5 FPGA on heterogeneous designs.

The next Section introduces the fault tolerance techniques using partial reconfiguration and the notation used in this paper. The Section III introduces the reliability level and the bitstreams sizes overhead evaluations, applied to the partial reconfiguration techniques. They are compared with real applications in Section IV. while Section V concludes the paper.

II. PARTIAL RECONFIGURATION MODES

An application is implemented on an FPGA by configuring a matrix of resources such as logic blocks, I/O, routing, memories and Digital Signal Processors (DSPs). This can be done in different ways, impacting the resources usage and the performances. Modern dynamically and partially reconfigurable FPGAs support two reconfiguration types [6]. In the 1D-reconfiguration type, the smallest addressable reconfigurable unit is a column that spans the whole height of the device. The 2D-reconfiguration type defines a reconfigurable area organized in columns and rows; the smallest addressable configurable unit is defined as a frame, which is a group of resources of the same type.

A tile is a group of frames that executes a function. An application mapped onto an FPGA, generally uses several tiles as illustrated in the Figure 1 by the square in bold. The functional area is defined by the configured frames (the "used" ones in blue in the figures), the other ones are the "spare frames" (in white in the figures).

In [5], the size of the tiles is considered as identical for all functions of the application. However, it is common to use differently-sized tiles depending on the functions needs in order to optimize the resources usage of the FPGA. The faults are seen as independent events following a Poisson

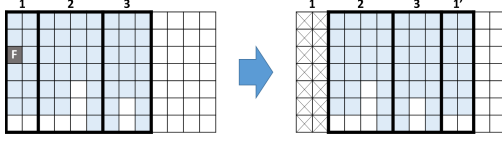


Fig. 1. Coarse redundancy reconfiguration.

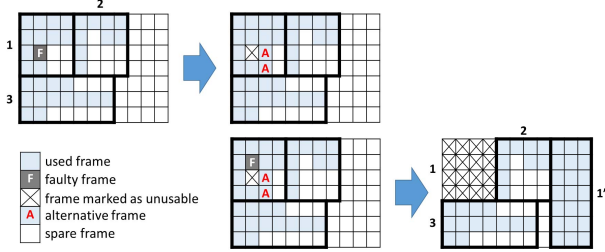


Fig. 2. Mixed-type reconfiguration.

distribution with a constant failure rate (λ_f) expressed in FIT^1 . The notations used hereafter are given in Table I. The reliability of a frame in an FPGA is obtained by:

$$r_f = \exp(-\lambda_f \cdot T_{\text{mission}} \cdot 10^{-9}) \quad (1)$$

Different reconfiguration techniques are defined depending on the partial reconfiguration abilities of the FPGA:

a) **Tile-based** [7]: the FPGA is partitioned into tiles. A faulty frame can be replaced by the spare frames from the same tile (upper part of Fig. 2).

b) **Coarse redundancy** [8]: this method can be applied to devices with 1D-reconfiguration type. The FPGA is then partitioned into columns-height tiles. The reliability is achieved at the coarse level, i.e. a faulty tile is replaced by a spare tile. In Figure 1, when a fault occurs in the tile 1, it is marked as unusable; then it is replaced by a spare one 1'.

c) **Mixed** [7]: is a combination of the two previous methods (Figure 2). If only one frame is faulty in a tile, the tile-based method is applied; if more than one fault occurs, then a spare tile is used (coarse redundancy).

d) **Naive-model**: is the same as the tile-based approach but with only one tile spreading out the whole FPGA. There is no need to partition the FPGA into tiles but it offers less flexibility.

III. EVALUATION OF THE RELIABILITY LEVEL

We propose here a generalization of the reliability evaluation formulas proposed in [5] to evaluate the worst-case scenario (i.e. the case of differently-sized tiles). Additionally, we introduce the estimation of the bitstream size overhead for each of the four reconfiguration techniques presented above.

A. Generic Evaluation of the Fault Tolerance Level

The reliability of a tile i is given by the Equation 2 (the size of the tile is n_{fpt_i}) with $x_{\text{tol_tile}}$ a binary parameter indicating if

¹one FIT is equal to one failure in 10^9 device-hours of operation

TABLE I
LIST OF THE NOTATIONS USED IN THE FORMULAS.

Notation	Definition
m	The total number of frames in an FPGA
λ_f	The failure rate per frame
T_{mission}	The mission duration in hours
r_f	The reliability of a frame
N	The number of tiles in an FPGA
N_{use}	The number of used tiles in the FPGA
n_{fpt_i}	The number of frames in a tile i
K_{t_i}	The number of used frames in a tile i
b_{pf}	The number of bits per frame
$N_{\text{tolCoarse}}$	Number of Tile supporting the coarse approach
R_{tile_i}	The reliability level of the tile i
R_{fpga}	The reliability level of the FPGA
B_{fpga}	The bitstream size required for the FPGA (bytes)

a fault tolerance methods is applied to the tile (0: no tolerance; 1: fault-tolerance), depending on the chosen technique.

$$R_{\text{tile}_i} = r_f^{K_{t_i}} + \left(K_{t_i} \cdot r_f^{K_{t_i}-1} \cdot (1 - r_f) \right) \cdot (x_{\text{tol_tile}} \cdot r_f^{n_{\text{fpt}_i} - K_{t_i}}) \quad (2)$$

At the inside-tile level, if $x_{\text{tol_tile}} = 1$, the worst-case scenario is to consider that a faulty frame is repaired by using all spare resources of the tile. The overall reliability level of the FPGA is obtained by the Equation 3, $x_{\text{tol_coarse}} = 1$ means that the fault tolerance at coarse level is applied, $x_{\text{tol_coarse}} = 0$ otherwise.

$$R_{\text{fpga}} = \left(\prod_{i=1}^{N_{\text{use}}} R_{\text{tile}_i} \right) + \left(\sum_{j=1}^{N_{\text{tolCoarse}}} \frac{\prod_{l=1}^{N_{\text{use}}} R_{\text{tile}_l} (1 - R_{\text{tile}_j})}{R_{\text{tile}_j}} \right) \cdot (x_{\text{tol_coarse}} \cdot r_f^{m - \sum_{i=1}^{N_{\text{use}}} n_{\text{fpt}_i}}) \quad (3)$$

B. General Bitstream size Evaluation

The Equation 4 calculates the total bitstream size overhead (i.e. bitstream including the alternative configurations) whether a reconfiguration technique is applied or not. The values of the parameters ϵ , η , τ depend on which fault tolerance strategy is applied; their values are detailed in Section III-C; $x_{\text{tol}} = 0$ for no tolerance technique applied, 1 otherwise.

$$B_{\text{tol}}(x_{\text{tol}}, \epsilon, \eta, \tau) = x_{\text{tol}} \cdot (\epsilon - \eta) \cdot \tau \cdot b_{\text{pf}} \quad \text{where: } \epsilon, \eta, \tau \in \mathbb{N} \quad (4)$$

The Figure 3 shows an example of pre-synthesized versions for the best case (one single spare resource is used) and the worst case (all the spare resources are used) of the tile-based technique. In the best case (from the reliability point-of-view), 9 versions have to be pre-synthesized (3 for each used frame). Meanwhile, there are 3 pre-synthesized versions in the worst case. Therefore, in the best case, the bitstream size overhead is $9 \times 6 \times b_{\text{pf}}$ bits, while in the worst case, it corresponds to $3 \times 6 \times b_{\text{pf}}$ bits.

At the inside-tile level, the worst case requires all spare resources to replace a faulty-frame. Thus, with K_{t_i} used frames, the bitstream overhead can be estimated using the Equation 4 with $(x_{\text{tol}}, \epsilon, \eta, \tau) = (x_{\text{tol}}, K_{t_i}, 0, K_{t_i} \cdot n_{\text{fpt}_i})$. In the best case, each spare frame can replace any faulty frame, therefore, with K_{t_i} used frames and $n_{\text{fpt}_i} - K_{t_i}$ spare frames,

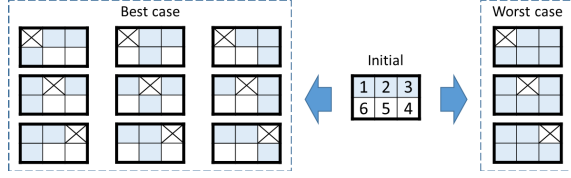


Fig. 3. Tile-based concept with the pre-configurations in the best and the worst cases.

the additional size of bitstream for the best case is obtained with $(x_{\text{tol}}, \epsilon, \eta, \tau) = (x_{\text{tol}}, n_{\text{fpt}_i}, K_{t_i}, K_{t_i} \cdot n_{\text{fpt}_i})$. Consequently, the bitstream size overhead corresponding to a single tile (for the best and the worst cases) can be obtained by the Equation 5.

$$B_{\text{tol_tile}} = \begin{cases} B_{\text{tol}}(1, K_{t_i}, 0, K_{t_i} \cdot n_{\text{fpt}_i}), & \text{the worst case} \\ B_{\text{tol}}(1, n_{\text{fpt}_i}, K_{t_i}, K_{t_i} \cdot n_{\text{fpt}_i}), & \text{the best case} \end{cases} \quad (5)$$

At the coarse-level, a tile marked as faulty is replaced by spare tile resources; the reconfiguration spans to the whole FPGA. In the worst case (from the reliability point-of-view) a tile must be replaced by the whole spare tiles. At the coarse level in the best case, a tile can be replaced by any single spare tile, while in the worst-case all spare resources are used for repair the faulty tile. The bitstream size overhead at the coarse level is estimated with $(x_{\text{tol}}, \epsilon, \eta, \tau)$ as defined in the Equation 6.

$$B_{\text{tol_coarse}} = \begin{cases} B_{\text{tol}}(1, N_{\text{tolCoarse}}, 0, m), & \text{the worst case} \\ B_{\text{tol}}(1, N, N_{\text{use}}, m \cdot N_{\text{tolCoarse}}), & \text{the best case} \end{cases} \quad (6)$$

The overall size of the bitstreams for the whole FPGA is defined by:

$$B_{\text{fpga}} = \sum_{i=1}^{N_{\text{use}}} \left(n_{\text{fpt}_i} \cdot b_{\text{pf}} + B_{\text{tol_tile}} \right) + B_{\text{tol_coarse}} \quad (7)$$

C. Application to the Fault-Tolerance methods

a) *Tile-based*: The reliability level of a tile-based reconfiguration technique can be directly estimated from R_{tile} (the Equation 2). For the tiles with fault tolerance enabled, the value of $x_{\text{tol_tile}} = 1$ and $B_{\text{tol_tile}}$ is calculated by the Equation 5. For the tiles with no fault tolerance, their $x_{\text{tol_tile}} = 0$ and $B_{\text{tol_tile}} = 0$. The overall reliability of the FPGA is calculated by the Equation 3 with $x_{\text{tol_coarse}} = 0$. The size of the bitstream is obtained by evaluating the Equation 7 with $B_{\text{tol_coarse}} = 0$.

b) *Coarse Redundancy*: This technique doesn't rely on the inside-tile level fault tolerance, thus both $x_{\text{tol_tile}}$ and $B_{\text{tol_tile}}$ are equals to 0. Meanwhile, the tiles in the set support the fault tolerance at the coarse level. Hence, the overall reliability can be obtained by the Equation 3 with $x_{\text{tol_coarse}} = 1$. The bitstream size overhead of this technique is estimated by the Equation 7 ($B_{\text{tol_coarse}}$ from Equation 6).

c) *Mixed*: The mixed technique combines the tile-based and the Coarse redundancy approaches. Therefore, for tiles with fault tolerance enabled, $x_{\text{tol_tile}} = 1$ and $B_{\text{tol_tile}}$ of the Equation 5 is considered. For the tiles without the fault

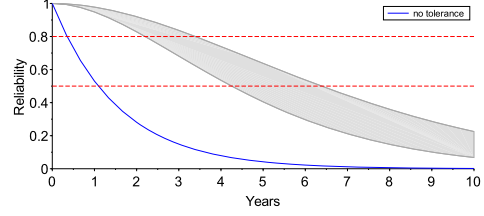


Fig. 4. The reliability level over years, for a mixed approach and identically-sized tile. The gray zone represents the gap between the best-case approach and the worst-case one.

tolerance applied, their $x_{\text{tol_tile}}$ and $B_{\text{tol_tile}}$ are set to 0. The overall reliability level can be estimated by the Equation 3 where $x_{\text{tol_coarse}} = 1$. The bitstream size overhead is then given by the Equation 7 using the $B_{\text{tol_coarse}}$ of the Equation 6.

d) *Naive-model*: As mentioned above, this technique is equivalent to the tile-based approach with only one tile occupying the whole FPGA, thus we can set $N_{\text{use}} = N_{\text{tolTile}} = 1$. The overall reliability of this technique is given by the Equation 2 where $x_{\text{tol_tile}} = 1$, $n_{\text{fpt}_1} = m$ and K_{t_1} is the number of used frames on the whole FPGA device. The overall size of bitstream is evaluated by the Equation 7 where $B_{\text{tol_coarse}} = 0$ and $B_{\text{tol_tile}}$ from the Equation 8:

$$B_{\text{tol_tile}} = \begin{cases} B_{\text{tol}}(1, K_{t_1}, 0, m), & \text{the worst case} \\ B_{\text{tol}}(1, m, K_{t_1}, K_{t_1} \cdot m), & \text{the best case} \end{cases} \quad (8)$$

IV. COMPARISONS OF APPROACHES

For sake of comparisons of our results with the previous work presented in [5] we used a Virtex 5 LX50 FPGA. This chip comprises $m = 9564$ frames (with $b_{\text{pf}} = 1312$ bits/frame). We assume a failure rate of 1 failure per year while in space environment 0.49 is usually considered [9]. The failure rate per frame is $\lambda_f = \frac{1 \cdot \text{failure-per-year}}{9564} = 1.04 \cdot 10^{-4}$ failure-per-year = 12 FIT since the frames are independant. The percentage of spare resources is set to $\sim 20\%$ according to [5]. This value has been chosen for comparison purpose but it actually highly depends on the design implementation.

A. Identical-Size Tiles

The reliability level and the bitstream size overhead for identical-size tiles partitioned FPGA are evaluated here. If all fault-tolerance techniques provide an improvement in terms of reliability level, the tile-based shows a superior advantage compared to the others. The mixed approach has an absolute advantage since the fault tolerance is applied at two levels (inside-tile and coarse redundancy). The Figure 4 illustrates the gap in terms of mission time between the best-case (7 years) from [5], and the worst-case (4.2 years) evaluation for a reliability level fixed to 0.5 for the mixed approach. The Table II gives the size of the overhead (in bits) required for each technique compared to a non-tolerant design. It appears that the mixed approach provides a good compromise in terms of improvement of the reliability level and bitstream size overhead. It can be noticed that the best-case for the reliability, becomes the worst-case in terms of bitstream size overhead.

TABLE II
BITSTREAM SIZE OVERHEAD OVER A NON-TOLERANT DESIGN FOR IDENTICAL-SIZE TILES FOR THE RECONFIGURATION TECHNIQUES.

(Gbit)	Tile-based	Coarse redundancy	Mix	Naive-model
Worst case	2.39	0.13	2.5	76.8
Best case	229.98	0.50	230.7	264,443

B. Different-size tiles

In real applications the FPGA is partitioned into different-size tiles. For our experiments, we used the system described in [9] which consists in six spatial applications. The Table III lists the applications and shows the resources attributed to each of them.

Without loss of generality we assumed here the the designer chooses to apply the tolerance-level reconfiguration following the rules:

- 80% frames in each tile are used, 20% of spare;
- for the tile-based technique: the tiles (3, 5, 6) apply it;
- for the coarse level: the tiles (3, 5, 6) apply this tolerance level;
- for the mixed strategy: tiles (3, 5, 6) apply the inside-tile tolerance level, all tiles apply the coarse tolerance level.

The evaluation from [5] cannot be evaluated since the authors considered homogeneous tile size only.

TABLE III
FPGA TILES PARTITION FOR THE SPACE APPLICATIONS.

Tile	1	2	3	4	5	6
Name	SpaceWire	Wavelet	AC97	MEPG4	8086	Ethernet
Frames	140	247	437	680	1068	2100

The Figure 5 illustrates the evaluation of the reliability level of the FPGA in terms of mission time. We can easily see that the mixed approach always prevails over the others due to the two-level ability of reconfiguration. The obtained reliability level always overcomes the naive-model when the tile-based and the coarse redundancy are applied, confirming the results obtained with the identical-size tiles. These techniques can tolerate one faulty frame per tile at the inside-tile level and one faulty tile at the coarse-level. The bitstream size overhead

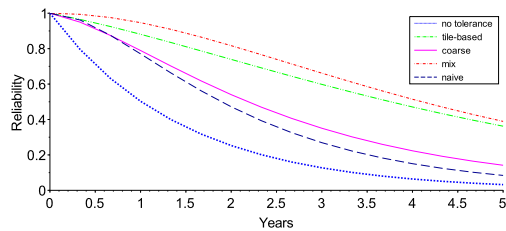


Fig. 5. Reliability level over years for different-size tiles for the different fault tolerant techniques.

due to the application of a reliability technique compared to the non-tolerant version of the system (i.e. including the applications) is 6.02 Gbits for the tile-based-approach, 0.38 Gbits

for the coarse redundancy approach, 6.07 Gbits for the mixed one and 58 Gbits for the naive-model. The naive-model is not the best candidate due to its bitstream size overhead and its limited performances. This is due to the very large amount of available spare frames needed to replace one faulty frame since all applications are placed into one tile.

The coarse redundancy needs the least pre-synthesized resources because replacing a tile completely by a spare tile is simpler than dealing with individual frames. Finally, the results show that the tile-based and the coarse redundancy are more suitable than the naive-model for applications with completely independent modules.

V. CONCLUSION

This work proposes an approach to evaluate the bitstream size overhead and the bounds of the reliability level for FPGA using partial reconfiguration for fault tolerance – tile-based, coarse, mixed and naive. The mixed approach offers the highest reliability level improvement while the coarse redundancy offers the smallest bitstream size overhead. The tile-based can be used with an independent-module application that allows each module to be easily configured inside their own tile. The coarse redundancy seems easiest to use when a tile can be completely replaced by a spare tile in an independent-module application (or on a specific FPGA). The mixed approach gives designers a great reliability improvement, but the complexity is higher since the fault tolerance is set at two levels (inside-tile and coarse). Besides, the naive-model is not really suitable due to its poor performances and its bitstream size overhead.

ACKNOWLEDGMENT

This work is supported by the Pays-de-la-Loire Region under the Holistar project funding.

REFERENCES

- [1] H. Zhang, L. Bauer, M. A. Kochte, E. Schneider, H.-J. Wunderlich, and J. Henkel, "Aging resilience and fault tolerance in runtime reconfigurable architectures," *IEEE Transactions on Computers*, vol. 66, no. 6, pp. 957–970, 2017.
- [2] H. A. Almurib, T. N. Kumar, and F. Lombardi, "Scalable application-dependent diagnosis of interconnects of sram-based fpgas," *IEEE Transactions on Computers*, vol. 63, no. 6, pp. 1540–1550, 2014.
- [3] A. Stoddard, A. Gruwell, P. Zabriskie, and M. J. Wirthlin, "A hybrid approach to fpga configuration scrubbing," *IEEE Transactions on Nuclear Science*, vol. 64, no. 1, pp. 497–503, Jan 2017.
- [4] B. Harikrishna and S. Ravi, "A survey on fault tolerance in fpgas," in *Intelligent Systems and Control (ISCO), 2013 7th International Conference on*. IEEE, 2013, pp. 265–270.
- [5] S. Pontarelli, M. Ottavi, V. Vankamamidi, G. C. Cardarilli, F. Lombardi, and A. Salsano, "Analysis and evaluations of reliability of reconfigurable fpgas," *Journal of Electronic Testing*, vol. 24, no. 1-3, pp. 105–116, 2008.
- [6] C. Bolchini, A. Miele, and C. Sandionigi, "A novel design methodology for implementing reliability-aware systems on sram-based fpgas," *IEEE Transactions on Computers*, vol. 60, no. 12, pp. 1744–1758, 2011.
- [7] M. Psarakis and A. Apostolakis, "Fault tolerant fpga processor based on runtime reconfigurable modules," in *Test Symposium (ETS), 2012 17th IEEE European*. IEEE, 2012, pp. 1–6.
- [8] S. Di Carlo, G. Gambardella, P. Prinetto, D. Rolfo, P. Trotta, and A. Vallerio, "A novel methodology to increase fault tolerance in autonomous fpga-based systems," in *On-Line Testing Symposium (IOLTS), 2014 IEEE 20th International*. IEEE, 2014, pp. 87–92.
- [9] J. Zhang, Y. Guan, and C. Mao, "Optimal partial reconfiguration for permanent fault recovery on sram-based fpgas in space mission," *Advances in Mechanical Engineering*, vol. 5, p. 783673, 2013.