Exploiting assertions mining and fault analysis to guide RTL-level approximation

Alberto Bosio*, Samuele Germiniani†, Graziano Pravadelli‡ and Marcello Traiola§
University of Verona, Department of Computer Science {† ‡}
Univ Lyon, ECL, INSA Lyon, CNRS, UCBL, CPE Lyon, INL, UMR5270 {*}
University of Rennes, Inria, CNRS, IRISA, UMR6074 {§}
Email: *alberto.bosio@ec-lyon.fr, †samuele.germiniani@univr.it, ‡graziano.pravadelli@univr.it, §marcello.traiola@inria.fr

***Index Terms*—Approximate computing, Assertion-based verification, Assertion mining, Fault injection.**

I. Introduction

Approximate Computing (AxC) paradigm was introduced to achieve higher power efficiency, lower area and better performances w.r.t. a "classical" computing system at the cost of a degraded, but still acceptable, output accuracy [1]. AxC can be applied at several abstraction levels of a given computing system: from circuit to algorithm [1], leading to a wide design exploration space that quickly became the bottleneck for successfully deploying AxC. Indeed, the literature proposes many works to automatically trade-off between output accuracy and performances [2]. However, most of them lack the capability to identify resilient elements (e.g, HW component, HDL statements, etc.) of the design to be approximated. Consequently, exploring the design for AxC generally results in a long and tedious procedure. Existing approaches generate approximate variants of the Design Under Exploration (DUE). Every variant is then executed/simulated in order to determine the accuracy degradation [3], which depends on the application and requires a specific metric to be computed (e.g., similarity index, hamming distance, etc.).

To help in this process, we propose an effective way for guiding the approximation of the DUE and identifying resilient elements at RTL, by leveraging the combination of fault injection and assertion-based verification.

II. Main idea

The methodology is composed of three phases: 1) Two approximation techniques are considered, bit-width reduction and statement reduction. Fault injection is used to mimic their effect on the DUE, obtaining a corresponding set of faulty DUEs. 2) Assertions are then automatically generated using the HARM assertion miner [4], [5] from the execution traces of the original DUE, capturing its golden functionalities. 3) The assertions are re-evaluated on each faulty DUE to analyse the effect of the fault, mimicking the approximation, with respect to the original DUE. This is done by comparing the contingency tables of the assertions evaluated on the golden and faulty traces. Finally, these variations are used to rank the different approximation alternatives according to their estimated impact on the functionality of the DUE.

The output of our methodology is a list of DUE elements (either bits of signals/registers or statements) ordered by increasing levels of criticality: the higher the criticality, the more prominent the effect of approximating an element on the functional correctness of the original DUE. A clustering procedure is finally exploited to determine which approximations should be applied simultaneously.

III. Case Study

We evaluated the proposed approach on the RTL description of a common Sobel edge-detection filter. We applied both the bit-width and the statement reduction approximation strategies on a set of approximation tokens (AT, i.e., bit tokens and statement tokens) by injecting faults. We then ranked and clustered the ATs by exploiting a set of automatically generated assertions. We finally measured the approximation effect on the designs synthesized by approximating the different ATs in terms of functional accuracy and power/area reduction. The goal is to show the effectiveness of our approach in prompting the designer to simultaneously approximate a cluster of ATs that has a low impact on functional accuracy while guaranteeing a relevant saving in terms of area and power. We considered a total number of 236 bit tokens and 38 statement tokens.

A. Functional accuracy

For evaluating the effectiveness of the proposed approach from the point of view of the functional accuracy of the approximated designs, we adopted the Structural SIMilarity (SSIM) index [6]. Eight images have been used as the workload for the Sobel. For the bit-with reduction, we make two different experiments: the first by fixing the target bit token to 0, and the second to 1. Since the outcome is similar for both scenarios, we report the results only for the stuck-at 0 case.

The diagrams in Figure 1(a)(b) show the SSIM (y axis) achieved by the designs obtained by applying the bit-width reduction (a) and the statement reduction (b) technique to each AT, i.e, each bit token in (a), and each statement token in (b). The x axis refers to the ATs ranked in decreasing order. There is a clear trend showing that the ATs that guarantee the highest SSIM, when approximated, are those ranked first by our approach, for both bit tokens and statement tokens.

However, since approximating a single AT is not effective from the point of view of area and power saving, we propose to cluster the ATs for applying multiple approximations simulta-

978-3-9819263-7-8/DATE23/© 2023 EDAA

neously. Thus, the diagrams in Figure 1(c)(d) present the effect of simultaneously applying the approximations belonging to the AT clusters using the k-means algorithm. The points in the dark line refer to the size of the clusters, the points in the blue line indicate the SSIM of the clusters returned by our methodology, and the points in the red line highlight the SSIM achieved by a set of ATs randomly selected, whose size is the same of the corresponding cluster in the blue line. A first observation highlights that the random clusters achieve the worst accuracy. The second observation deals with the quality of the ranked clusters returned by our methodology. The cluster containing the top-ranked ATs for the statement reduction strategy (Figure 1 (d)) shows that simultaneously applying the approximations of the top-ranked cluster (ID 0) guarantees almost the best SSIM; however, this is not the same for the bit-width reduction strategy (Figure 1 (c)), where larger clusters (e.g., ID 0) achieve an unsatisfactory SSIM, while smaller ones (e.g., IDs 1 and 2) generally perform better. That demonstrates that the SSIM for the bit-with reduction strategy is influenced by the size of the cluster, whereas it is not relevant for the statement reduction strategy. In addition, by analysing the composition of the clusters containing the statement tokens versus those related to the bit tokens, we observed that the former generally collects ATs belonging to the same cone-of-logic, while this is not true for the latter. As a consequence, the impact of the approximation on the functional accuracy is amplified when a large set of unrelated bit tokens are clustered; conversely, this effect is mitigated while grouping statement tokens belonging to the same cone-of-logic. Therefore, we conclude that the ranking and clustering procedure for statement tokens is effective for guiding the exploration of the design using the statement reduction approximation strategy. In the next section, we show that this approach performs well also in terms of area/power savings.

B. Area and power saving

We performed the synthesis of the designs obtained by approximating the statement tokens as indicated in the clusters obtained with the proposed methodology. We targeted the FreePDK45 45-nm standard cell technology library. Figure 2 reports the results, in terms of relative area and power consumption, calculated as $1 - \frac{Precise - clus_i}{Precise}$ (*Precise* variant has value 1). The graph highlights that the cluster with ID 0 (the best ranked by our methodology) shows the best area and power reduction w.r.t. the *Precise* design, i.e., 54.53% and 50.24% respectively, while still having a high SSIM metric value, i.e. 0.62, as reported in Figure 1(d). Conversely, approximating a random set of statements with the same size as cluster 0 (i.e., 14 statements) provides a design having, on average, 58.69% area reduction and 30.60% power reduction; however, presenting also a lower SSIM value, i.e. 0.061. The results for the *clus_rnd* design were computed by synthesizing multiple designs obtained by randomly approximating the same number of statements as in cluster 0; finally, the average value is reported. Overall, the saving in terms of area and power achieved by approximating the clusters of statements identified by the proposed approach is generally proportional to the functional accuracy: the higher the saving, the higher the accuracy.

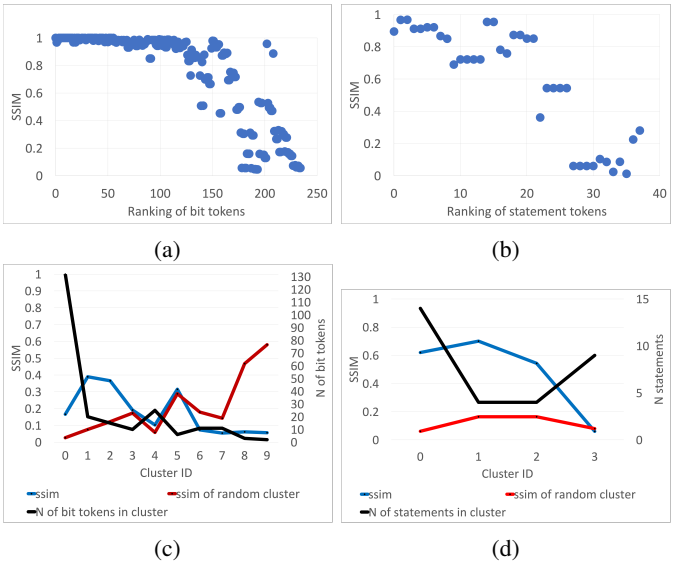

Figure 1: Impact on the Sobel functionality by AT approximation alternatives (a and b), and by simultaneously applying the approximations belonging to each AT clusters (c and d).

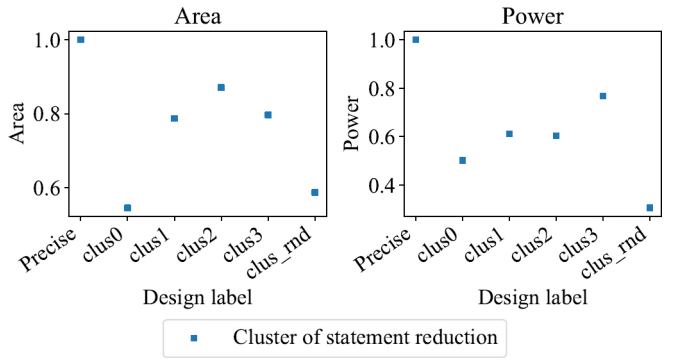

Figure 2: Saving in terms of area and power by considering the statement token clusters. *Precise* refers to the original design; *clus0*, *clus1*, *clus2* and *clus3* are related to the designs approximated according to the four clusters returned by the our methodology in decreasing order of functional accuracy; *clus_rnd* indicates the average result for the design approximated by using a set of randomly chosen ATs.

References

[1] A. Bosio *et al.*, Eds., *Approximate computing techniques*, 1st ed. Cham, Switzerland: Springer Nature, Jun. 2022.

[2] S. Mittal, "A survey of techniques for approximate computing," *ACM Comput. Surv.*, vol. 48, no. 4, pp. 62:1–62:33, Mar. 2016.

[3] S. Barone *et al.*, "Multi-objective application-driven approximate design method," *IEEE Access*, vol. 9, pp. 86 975–86 993, 2021.

[4] S. Germiniani *et al.*, "Harm: A hint-based assertion miner," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 11, pp. 4277–4288, 2022.

[5] https://github.com/SamueleGerminiani/harm.

[6] Zhou Wang *et al.*, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.