Investigation of Mean-Error Metrics for Testing Approximate Integrated Circuits

Marcello Traiola, Arnaud Virazel, Patrick Girard, Mario Barbarcschi, Alberto Bosio

Investigation of Mean-Error Metrics for Testing Approximate Integrated Circuits

Marcello Traiola[1], Arnaud Virazel[1], Patrick Girard[1], Mario Barbareschi[2], Alberto Bosio[1]
[1]LIRMM - University of Montpellier / CNRS - France - Email: {firstname.lastname}@lirmm.fr
[2]DIETI - University of Naples Federico II - Italy - Email: mario.barbareschi@unina.it

***Abstract*—Approximate Computing (AxC) is increasingly becoming a new design paradigm for energy-efficient Integrated Circuits (ICs). Specifically, application resiliency allows a trade-off between accuracy and efficiency (energy/area/performance). Therefore, in recent years, Error Metrics have been proposed to model and quantify such accuracy reduction. In addition, Error thresholds are usually provided for defining the maximum allowed accuracy reduction. From a testing point of view, Approximate Integrated Circuits offer several opportunities. Indeed, approximation allows one to individuate a subset of tolerable faults, which are classified according to the adopted threshold. Thanks to fewer required test vectors, one achieves test-cost reduction and improvements in yield. Therefore, using metrics based on the calculation of Mean Errors (ME metrics), has become a major testing challenge. In this paper, we present this problem and investigate the technical requirements necessary for ME metric testing. We perform experiments on arithmetic circuits to study opportunities and challenges in terms of complexity. Our results show that one can filter up to 21% of faults and also highlight the complexity of the problem in terms of execution-time.**

Keywords: Approximate Computing; Testing; ATPG; Functional Approximation; Integrated Circuits

I. Introduction

Over the last few years, many research works proved that some computing domains are inherently resilient to inaccuracy. Although some inner operations, or involved data, of a computing system are inexact, some applications are able to produce good-enough results [1]–[4]. The Approximate Computing (AxC) paradigm benefits from such a property by providing gains in efficiency (i.e., less power consumption, less area, higher manufacturing yield) at the cost of a slight accuracy reduction. The inaccuracy can involve every system layer from hardware to software components [5]. In this paper we focus on *Functional Approximation* [1], [6]–[15] applied to hardware components. The Functional Approximation aims at modifying the circuit structure so that its original functionality is replaced by a similar one, whose implementation leads to an area/energy reduction at the cost of a reduced accuracy. This means that a variation can be observed between the output values of the original IC and those of the approximate integrated circuit (AxIC). Such variation is the accuracy loss measured by means of Error Metric(s). For instance, we can mention the Error Rate, i.e. how many times an error is observed at the circuit outputs, and the Error Magnitude, measured as the difference between the golden and erroneous outputs, both formally defined in [3].

During the manufacturing process, physical defects (either random or systematic) can affect the Integrated Circuit (IC) and may be the cause of faults leading to observable errors. These errors (due to faults) may further reduce the accuracy - already reduced as result of the functional approximation - and may affect outputs more than expected. In this context, the role of testing is to ensure that the observed error due to the presence of defects is never greater than the acceptable error threshold fixed by the final user. In other words, all the faults that reduce the circuit accuracy more than allowed must be tested. Authors of [16], presented a pre-process to classify each fault of the AxIC either as *approximation-redundant* (i.e., tolerable compared to the threshold) or as *non-redundant* (i.e., non-tolerable), before applying the classical Automatic Test Pattern Generation (ATPG). In a previous work [17], we presented an approximation-aware ATPG approach to generate test vectors only for non-tolerable faults. The above mentioned works consider only metrics based on local or maximum errors, such as:

- Worst Case Error (WCE): the largest possible error between the outputs of the precise and approximate circuits;
- Maximum Bit-flip error (MBFE): the largest possible hamming distance between the outputs of the precise and approximate circuits.

As far as we know, there are not any works that face the problem of testing Approximate Circuits considering more complex metrics, such as those which require the calculation of mean errors. Among them, we can include:

- Mean Absolute Error (MAE): the sum of all the Error Magnitudes, averaged over all the input vectors, where:
  - the *Error Magnitude* (EM) is the absolute difference between the precise and approximate circuit outputs;
- Mean Squared Error (MSE): the sum of all squared EMs, averaged over all the input vectors;
- Error Probability (EP): the percentage of incorrect outputs among all the possible outputs.

The fundamental problem related to the above mentioned metrics is their complexity in terms of number of input combinations related to their computation. Therefore, in this paper we present the problems related to testing AxICs considering this kind of metrics and we investigate challenges and opportunities.

The remainder of the paper is organized as follows. Section II describes the above mentioned issues. Section III describes an approach to deal with them. Experimental results

are presented in Section IV. Finally, conclusions and some future directions are given in Section V.

II. Problem Statement

As described in Section I, functional approximation modifies/simplifies the circuit structure by relaxing some design requirements at the cost of introducing a certain amount of error. During the manufacturing process, physical defects could cause an error greater than the acceptable one. Therefore, in this context, testing aims at avoiding that AxICs affected by unacceptable errors are shipped to the customer. The general and fundamental assumption is that only one fault at a time could occur within the circuit. This relies on the statistic that failures are only rarely the product of two or more simultaneous faults.

In general, given the list of all possible faults that can occur within an IC (whether approximate or not), each detectable fault impacts on the circuit outputs. By considering different metrics, the impact of such faults can be measured and expressed as error. Given a metric M, we can measure the error e_i induced by a fault f_s stimulated by the input vector i. By considering another metric $\widehat{M}$, the error due to the same fault f_s is measured as $\widehat{e_i} \neq e_i$ when stimulated by the same input vector i. Moreover, by stimulating the fault with two different input vectors i and j, the measured errors will be $e_i \neq e_j$ ($\widehat{e_i} \neq \widehat{e_j}$).

For clarifying the idea, let us consider as example two metrics for arithmetic circuits: the Error Magnitude (EM) (or *Arithmetic Distance*) and the Bit-Flip Error Metric (BFE). The first can be expressed as the absolute value of the arithmetic difference of two values. The second can be expressed as the hamming distance of two sequences of bits. As depicted in

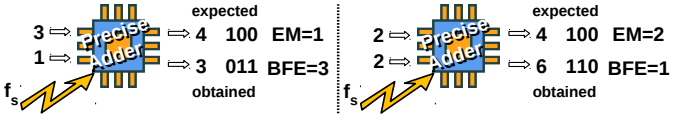


Fig. 1: Fault impact depending on metrics

Figure 1, the error measured at the output of a circuit as a consequence of a fault f_s depends on the input vectors, as well as the considered metric. Indeed, by stimulating the circuit with two different input vectors ([3,1],[2,2]), f_s induces the errors $e_i = 1$ and $e_j = 2$, measured by considering the EM metric; on the other hand, the errors $\widehat{e_i} = 3$ and $\widehat{e_j} = 1$ are measured by considering the BFE.

In the context of AxICs, the goal of the testing is to identify the whole set of detectable faults whose impact on the circuit outputs is non-acceptable compared with a chosen metric (i.e., the error is greater than the given threshold), for all the possible combinations of inputs.

Figure 2 represents the above concept. Once considered a specific metric and a threshold, the set of all possible faults which can affect an AxIC can be classified into two subsets, depending on the error E induced by the faults. We refer to F_T as the set of faults which would not induce an error greater than the given threshold. Conversely, we name F_S the

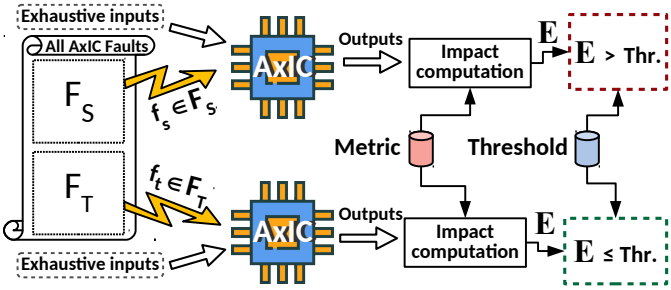


Fig. 2: AxIC Fault impact

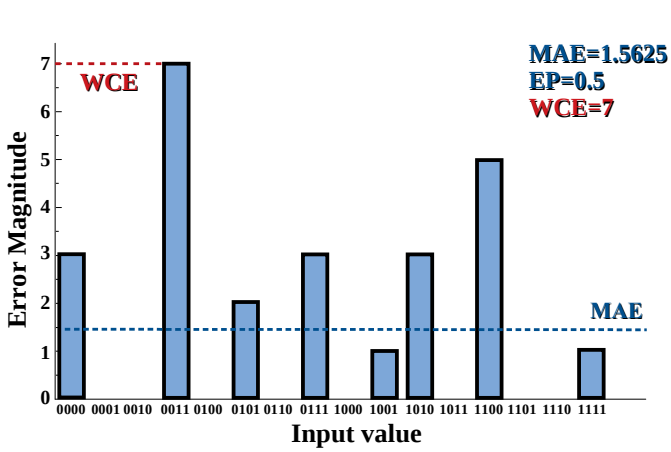
set of faults which would induce an error greater than the given threshold. Testing only for the set of detectable faults F_S guarantees to have an error that does not exceed the acceptable one, defined by the threshold.

The advantage of applying such procedure is, above all, the yield increment (i.e., fewer circuits will be rejected). Moreover, by reducing the number of faults to be tested, the size of the test set is also expected to be reduced. This results in lower test costs. The reduction of the test time is very important especially in the perspective of online testing.

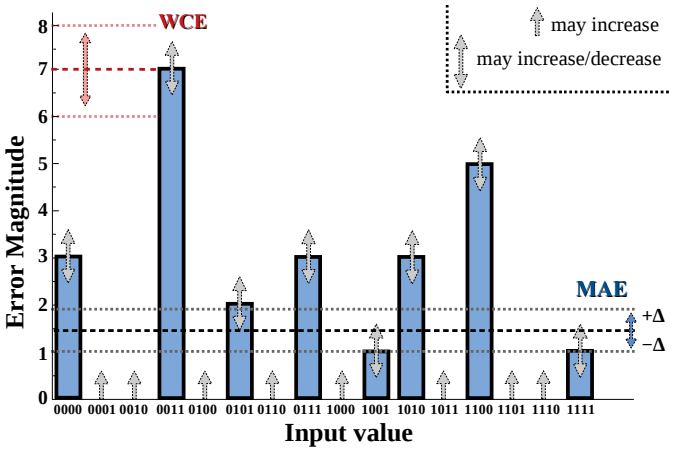
Depending on the considered metric, the complexity of the classification process can change significantly. In [16], authors stated that the problem of finding the so-called *approximation-redundant faults* - whose effect will always be below the given threshold - is #P-complete when the considered metric is the Worst Case Error. Conversely, finding approximation-redundant faults considering the Error Probability metric turns out to be a NP-complete problem. Moreover, in [17], we proposed an ATPG-based technique capable of generating test vectors only for non-redundant faults, classified according to the WCE metric. Such technique relies on the usage of a circuit that computes the error metric and evaluates it against the given error threshold. Thanks to that, for a given fault, the ATPG can quickly find an input vector producing an output affected by an error greater than the WCE. Thus, the fault is classified as *non-redundant*. If the ATPG cannot find an input vector, the fault is classified as *approximation-redundant*.

Unfortunately, this analysis is not sufficient to ensure that a fault will not impact on metrics which require the calculation of mean errors (Mean-Error Metrics or ME Metrics).

Let us resort to an example to depict the issue. We consider a 2-bits arithmetic circuit (that we call "original" circuit) to which we apply a functional approximation technique. The outcome is an approximate circuit that is more efficient (e.g. it has reduced area or reduced energy consumption or better performance) but shows some errors at outputs. The graph shown in Figure 3a represents the hypothetical *error magnitude profile* of such *fault-free* approximate arithmetic circuit (i.e. the circuit produces such errors due to the approximation and not due to manufacturing faults). By considering three different metrics - MAE (Equation 1), EP (Equation 2), and WCE (Equation 3) - the measured error changes ($MAE = 1.5625$, $EP = 0.5$, and $WCE = 7$).



(a) Error profile of a fault-free approximate arithmetic circuit



(b) Error profile of a faulty approximate arithmetic circuit

Fig. 3: Error profile of a fault-free approximate arithmetic circuit (a); error profile possible variations of the same approximate arithmetic circuit in presence of a fault (b).

$$\text{MAE} = \frac{\sum_{\forall i \in \mathcal{I}} \left| O_i^{\text{approx}} - O_i^{\text{orig}} \right|}{2^n} \tag{1}$$

$$\text{EP} = \frac{\sum_{\forall i \in \mathcal{I}:\ O_i^{\text{approx}} \neq O_i^{\text{orig}}} (1)}{2^n}. \tag{2}$$

$$\text{WCE} = \max_{\forall i} \left| O_i^{approx} - O_i^{orig} \right| \tag{3}$$

We refer to $\mathcal{I}$ as the set of all the possible input combinations and to n as the number of input bits. The i-th bar of the graph in Figure 3a reports the arithmetic distance (i.e., error magnitude) between the original output (i.e. the output of the original circuit) and the approximate output (i.e. the output of the approximate circuit), measured when applying the i-th input vector. This is exactly the value of $|O_i^{\text{approx}} - O_i^{\text{orig}}|$. The mean value over all the inputs is represented by the MAE, the WCE represents the maximum value, and the ratio of the number of bars to the numbers of inputs vectors gives the EP. We can now imagine that, during the manufacturing phase, a fault (i.e., Stuck-at-fault) is introduced within the circuit. Its impact on the MAE depends on the variation of each bar of the graph, as shown in Figure 3b. In other words, it depends on the error magnitude of the AxIC for each possible input. Similarly, the impact of the fault on the EP depends on the total number of input vectors which generate an error. The WCE value changes only if the maximum possible error changes, as a result of the fault. Thus, to perform the fault classification w.r.t. the WCE is sufficient prove either the existence or the non-existence of an input vector which increase the maximum possible error, for a given fault. Figure 4 and 5 illustrates the above consideration. Figure 4 depicts the hypothetical impact of a fault f_1 on the Error profile of the 2-bit arithmetic circuit mentioned above. The fault impacts the error magnitude when applying the input vectors "0011" and "1100". While the WCE is increased by 1, the MAE metric remains unchanged. Indeed,

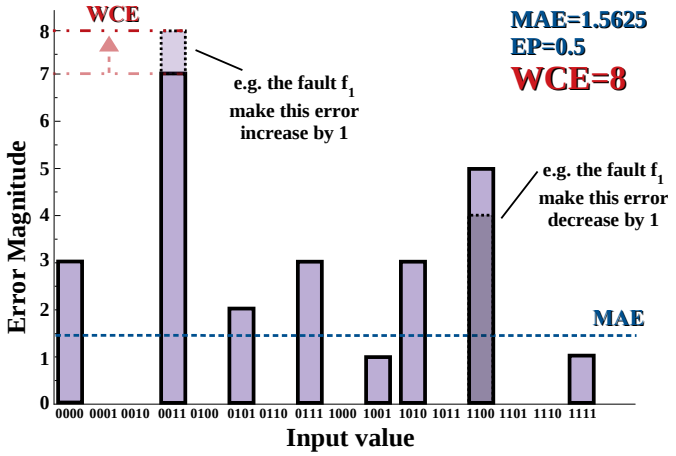


Fig. 4: Impact of the fault f_1: WCE increases, MAE and EP remain unchanged

an error magnitude increase of 1 is measured when applying the input vector "0011", modifying the WCE value from 7 to 8. By applying the vector "1100", the measured error magnitude is decreased by the same amount, leaving the MAE unchanged. Moreover, also Error Probability (EP) remains unchanged.
In the same way, Figure 5 describes the hypothetical impact of another fault (f_2) on the same circuit. In this case, the fault impact can only be measured by applying the input vector "1110". The measured error increased from 0 to 1. In this case, the WCE remains unchanged while both MAE and EP increase.
Finally, the two mentioned faults would be filtered or not, depending on the considered metric. Specifically, f_1 would be filtered only when considering WCE, whereas f_2 only when considering EP and/or MAE.

Hence, it is less complex to evaluate the impact of a fault when considering metrics which only need a single condition to be met. Indeed, for a given fault, if we prove the existence of a single vector that makes the error exceed the threshold, we can state that such fault is non-redundant w.r.t. the WCE.

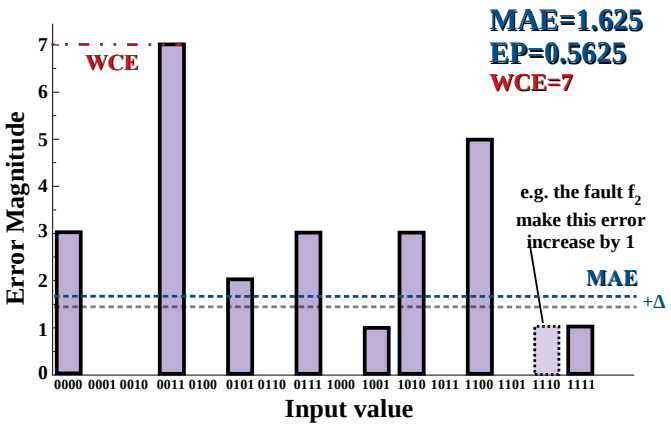


Fig. 5: Impact of the fault f_2: WCE remains unchanged, MAE and EP increase

Conversely, classifying faults w.r.t. ME metrics is a $O(2^n)$ complexity problem, with n = number of input bits. Indeed, to state whether a fault generates an error exceeding the threshold or not, we need to know the error contribution for all the input vectors. In the next section, we introduce an approach for filtering approximate-redundant faults considering ME metrics.

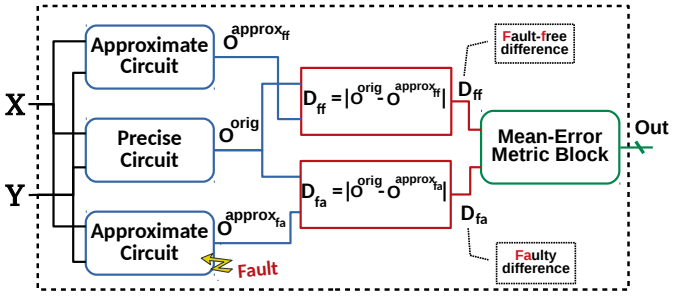
III. Mean-Error Metrics Aware Testing of AxICs

As stated in Section II, the process of classifying faults by considering ME metrics is not trivial. Therefore, we aim at studying the opportunities offered by such classification. To illustrate the underlying idea, let us consider the eqs. (1), (2) and (4), which formalize the ME metrics defined for arithmetic circuits [18].

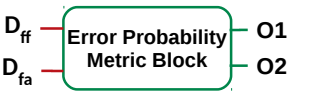
$$\text{MSE} = \frac{\sum_{\forall i \in \mathcal{I}} \left| O_i^{\text{approx}} - O_i^{\text{orig}} \right|^2}{2^n} \tag{4}$$

The impact of a fault within the approximate circuit affects the AxIC outputs (i.e., the O_i^{approx} value) for a subset of input combinations. As shown in Figure 3a we are interested in the error magnitude variation for all input combinations (i.e., the variation of $|O_i^{\text{approx}} - O_i^{\text{orig}}| \ \forall i \in \mathcal{I}$). The goal is to understand whether a fault impact increases or not the value of the sum of all the errors, for all input combinations (i.e., the term $\sum_{\forall i \in \mathcal{I}} \left| O_i^{\text{approx}} - O_i^{\text{orig}} \right|$). For the special case of EP, it is enough to study a fault impact on the number of input combinations which cause $O_i^{\text{approx}} \neq O_i^{\text{orig}}$.

For this reason, we propose a Fault Filtering Architecture (FFA), shown in Figure 6a. This circuit is never manufactured. It is only used to support the fault classification into approximation-redundant and non-redundant. Given the input vector [X,Y], the fault affecting the AxIC, and a specific ME metric, this architecture is capable to determine whether such fault changes or not the metric value, for that vector (i.e., a single bar in Figure 3a). The "Mean-Error Metric Block" will depend on the target metric. As depicted in Figure 6b, for EP metric we use a block with only two output bits whose values state whether the fault has changed (increased or decreased)

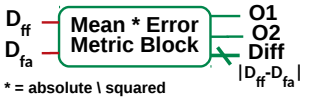


(a) Fault Filtering Architecture for ME metrics



O1	O2	Logic	Meaning
0	0	$D_{ff} \neq 0$ AND $D_{fa} \neq 0$	EP does not change
1	0	$D_{ff} \neq D_{fa}$ AND $D_{ff} \neq 0$	EP increases by 1
0	1	$D_{ff} \neq D_{fa}$ AND $D_{fa} \neq 0$	EP decreases by 1

(b) EP metric Block logic



O1	O2	Logic	Meaning
0	0	$D_{ff} = D_{fa}$	M*E does not change
1	0	$D_{ff} < D_{fa}$	M*E increases by Diff value
0	1	$D_{ff} > D_{fa}$	M*E decreases by Diff value

(c) M*E metric Block logic

Fig. 6: FFA

the value of the Error Probability or not, for the given input vector. The M*E (MAE and MSE) metric block, shown in Figure 6c, has an additional output signal that reports the metric value variation.

Let us take as example the MAE metric. By using the notation $D_i = |O_i^{\text{approx}} - O_i^{\text{orig}}|$, we apply MAE Equation (1) to fault-free (ff) and faulty (fa) AxICs in order to show the theory behind this approach:

$$MAE_{ff} = \frac{\sum_{\forall i \in \mathcal{I}} \left| O_i^{\text{approx}_{ff}} - O_i^{\text{orig}} \right|}{2^n} = \frac{\sum_{\forall i \in \mathcal{I}} D_{ff_i}}{2^n} \tag{5}$$

$$MAE_{fa} = \frac{\sum_{\forall i \in \mathcal{I}} \left| O_i^{\text{approx}_{fa}} - O_i^{\text{orig}} \right|}{2^n} = \frac{\sum_{\forall i \in \mathcal{I}} D_{fa_i}}{2^n} \tag{6}$$

$$\Delta MAE = MAE_{fa} - MAE_{ff} = \frac{\sum_{\forall i \in \mathcal{I}} D_{fa_i} - D_{ff_i}}{2^n} \tag{7}$$

As stated in Equation 5, the metric value for the fault-free AxIC can be expressed as the sum of the D_{ff} values for all the input combinations (see Figure 6a). In the same way, Equation 6 states that the value of the metric for the faulty AxIC can be expressed as the sum of the D_{fa} values for all the input combinations. Finally, Equation 7 represents the target value of the investigation: the variation of the metric value due to presence of the fault. If the ΔMAE value is less than or equal to zero, then the fault can be considered as *approximation-redundant* and filtered. Otherwise, the fault must be tested.

The same considerations can be applied to the MSE metric. In addition, for each of the M*E metrics, the Δ value is proportional to the variation of the sum $\sum_{\forall i \in \mathcal{I}} D_i$. This is the value obtained as output of the M*E metric Block after applying all the inputs $\in \mathcal{I}$. Thus, the number of faults that

will be filtered is exactly the same for the two metrics.

As for EP metric, let us introduce the following function:

$$u(D_i) = \begin{cases} 1, & \text{if } D_i > 0 \\ 0, & \text{if } D_i = 0 \end{cases} \quad \text{where } D_i = \left| O_i^{\text{approx}} - O_i^{\text{orig}} \right| \tag{8}$$

By leveraging Equation 8, we can apply EP Equation (2) to fault-free (ff) and faulty (fa) AxICs as follows:

$$EP_{ff} = \frac{\sum_{\forall i \in \mathcal{I}} u\left(D_{ff_i}\right)}{2^n} \tag{9}$$

$$EP_{fa} = \frac{\sum_{\forall i \in \mathcal{I}} u\left(D_{fa_i}\right)}{2^n} \tag{10}$$

$$\Delta EP = EP_{fa} - EP_{ff} = \frac{\sum_{\forall i \in \mathcal{I}} u\left(D_{fa_i}\right) - u\left(D_{ff_i}\right)}{2^n} \tag{11}$$

Just as Equation 7, if the ΔEP value is less than or equal to zero, then the fault can be considered as *approximation-redundant* and filtered. Otherwise, the fault must be tested.

Finally, by knowing the subset of input vectors $\mathcal{J} \subset \mathcal{I}$ that stimulate and propagate each fault, one can perform the classification. Indeed, by simulating vectors belonging to $\mathcal{J}$ while injecting - one by one - all the faults, allows us to accomplish the goal. However, we left out for future works the problem of characterizing the subset $\mathcal{J}$.

In this work, we applied the FFA-based technique, by applying the exhaustive set of input vectors. The simulation produces a detailed report about the fault impact on the error profile. Afterwards, we perform a report analysis for extracting the information about the fault impact on the metric being investigated. Figure 7 sketches the overall flow. By

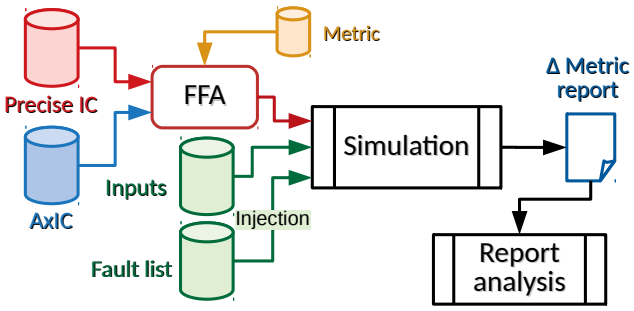


Fig. 7: Overall flow

applying the above described approach, we carried out some experiments. In the next section, we report the results. The goal is to investigate the opportunities offered by the classification of AxIC faults when considering metrics based on the mean errors. In other words, these experiments are not intended to prove the efficiency of the technique; rather, we want to asses the upper bound for the number of faults that can be filtered, when using ME metrics to measure the error.

IV. Experimental Results

In this section, we report experimental results obtained by applying the proposed approach on several approximate arithmetic circuits. Indeed, we applied the proposed approach on 448 non-dominated 8-bit approximate adders and 471 non-dominated 8-bit approximate multipliers taken from the EvoApprox8b library [18]. Adders were obtained by functional approximation of a Ripple-Carry Adder (RCA), a Carry-Select Adder (CSA), a Carry-Look-ahead Adder (CLA), a multiple Tree Adder (TA) and a Higher Valency Tree Adder (HVTA). As for multipliers, they were obtained by functional approximation of Ripple-Carry Array, multiple Carry-Save Array and Wallace Tree architectures. We synthesized the circuits using Synopsys Design Compiler and a 65-nm industrial CMOS technological library. We utilized the Fault Manager and the simulator within Synopsis TetraMAX to generate fault lists and perform simulations. Concerning the complexity, 8-bit adders are composed, on average, of 57 nodes (min 30, max 128); 8-bit multipliers are composed, on average, of 453 nodes (min. 239, max. 787).

We performed the experiments by evaluating EP metric and M*E metrics (i.e., MAE and MSE) and by considering the Stuck-at-fault model. In Figure 8 and Table I we report the results of the experiments. Specifically, for each metric and each circuit, we calculated the percentage of faults that do not induce on the circuit an error greater than the maximum allowed. Figure 8 depicts, for each circuit type, the percentage

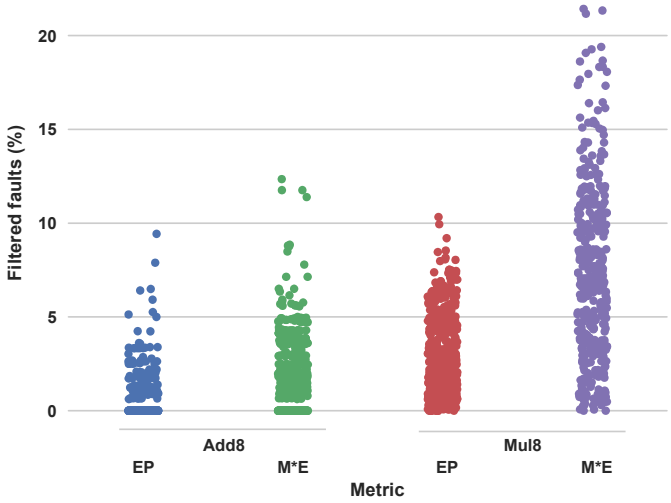


Fig. 8: Distribution of the percentage of filtered faults considering different metrics and circuits

of filtered faults for both the M*E metrics group and the EP metric. For each experiment group, the x-axis is spread out for clarity. For the metrics belonging to M*E group (MAE, MSE), we performed the very same analysis. Therefore, the number of filterable faults is the same. In Table I the *five-number summary* is reported (i.e., the five most important sample percentiles). In details, the Min column reports the sample minimum (the smallest amount of filtered faults), the Q1 column the first quartile, the Med column the median (the middle value), the Q3 column the third quartile and the Max column the sample maximum (the largest amount of filtered faults). In addition, Avg column reports the average

Circuits	Metric	Min	Q1	Med	Q3	Max	Avg	AvgTime(s)
Adders	**EP**	0.00%	0.00%	0.00%	0.90%	9.43%	**0.59%**	**106.73**
	M*E	0.00%	0.00%	1.08%	2.97%	12.35%	**1.83%**	**448.05**
Multipliers	**EP**	0.00%	0.94%	2.16%	4.35%	10.33%	**2.85%**	**924.50**
	M*E	0%	3.67%	6.72%	10.23%	21.42%	**7.22%**	**72164.9**

TABLE I: Experiment results - five-number summary

value and the last column the average time, in seconds, to analyze a single circuit. The fault reduction is calculated as the percentage of filtered faults over the total number of faults: $\text{Fault Reduction} = \frac{\text{Approximate-redundant faults}}{\text{Total Faults}} * 100$

Results show that it is possible to filter up to 9% of the faults for the 8-bit adders, when considering EP metric and up to 12% in the case of M*E metrics. In the case of Multipliers we were able to filter up to 10% of the faults when analyzing the EP metric and up to 21% when evaluating M*E metrics. However, comparing results with previous works, we can notice that the fault reduction gives better results when evaluating errors using the WCE metric. Indeed, in [17], 42% of faults were filtered on average, for 8-bit adders (18% min, 99% max). Concerning 8-bit multipliers, 59% of faults were filtered on average (5% min, 85% max). In [16], authors filtered on average 53% of the faults, by applying their methodology to arithmetic circuits. In the collected experiments for ME metrics, the average of filtered faults is not that promising: for 8-bit adders, only 0.59%, by analyzing EP metric and 1.83%, when evaluating M*E metrics; multipliers gave slight better results: 2.85%, when considering EP and 7.22%, by measuring the M*E. Ultimately, the opportunity of filtering faults by considering ME metrics appears not so attractive if compared to the required effort. Indeed, due to the complexity of the problem, the average time we needed to realize the experiments was very high, as shown in Table I. Nevertheless, this work allows to define the upper bound of faults that can be filtered within AxICs, when considering ME metrics. Consequently, this opens to further research for finding more efficient methodology to reduce the problem complexity and thus the execution time.

V. Conclusions

In this paper, we presented the problems related to the test of approximate digital circuits considering Error Probability (EP) and Mean-Error (ME) metrics, such as Mean Absolute Error (MAE) and Mean Squared Error (MSE). The core problem is to ensure that the faults introduced in the manufacturing phase do not introduce errors greater than the acceptable error threshold. From this perspective, we are allowed to filter *approximate-redundant* faults and generate test vectors only for faults which impact negatively the considered metric. Since the above mentioned metrics are strictly related to all the possible combinations of the circuit inputs, the problem is not trivial. For this purpose, we proposed a methodology to investigate the possible opportunities of classifying faults considering ME metrics. As far as we know, this is the first attempt to address such problem. We proposed a Fault Filtering Architecture and we performed experiments on several 8-bit approximate arithmetic circuits to assess the upper bound of faults that can be filtered. Results showed that it is possible to filter up to 21% of the faults. On the other hand, given the high complexity of the problem, the required time to apply the methodology is very long. Therefore, in the future, we aim to reduce the problem's complexity by reducing the number of input signals needed for calculating the metric variation.

References

[1] S. Mittal, "A survey of techniques for approximate computing," *ACM Comput. Surv.*, vol. 48, no. 4, pp. 62:1–62:33, Mar. 2016. [Online]. Available: http://doi.acm.org/10.1145/2893356

[2] Q. Xu, T. Mytkowicz, and N. S. Kim, "Approximate computing: A survey," *IEEE Design Test*, vol. 33, no. 1, pp. 8–22, Feb 2016.

[3] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *2013 18th IEEE European Test Symposium (ETS)*, May 2013, pp. 1–6.

[4] V. K. Chippa, S. T. Chakradhar, K. Roy, and A. Raghunathan, "Analysis and characterization of inherent application resilience for approximate computing," in *Proceedings of the 50th Annual Design Automation Conference*. ACM, 2013, p. 113.

[5] V. K. Chippa, S. Venkataramani, S. T. Chakradhar, K. Roy, and A. Raghunathan, "Approximate computing: An integrated hardware approach," in *Asilomar Conference on Signals, Systems and Computers*. IEEE, 2013, pp. 111–117.

[6] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," *IEEE Transactions on Computers*, vol. 64, no. 4, pp. 984–994, April 2015.

[7] P. Kulkarni, P. Gupta, and M. Ercegovac, "Trading accuracy for power with an underdesigned multiplier architecture," in *2011 24th Internatioal Conference on VLSI Design*, Jan 2011, pp. 346–351.

[8] D. Shin and S. K. Gupta, "Approximate logic synthesis for error tolerant applications," in *2010 Design, Automation Test in Europe Conference Exhibition (DATE 2010)*, March 2010, pp. 957–960.

[9] S. Venkataramani, A. Sabne, V. Kozhikkottu, K. Roy, and A. Raghunathan, "Salsa: Systematic logic synthesis of approximate circuits," in *DAC Design Automation Conference 2012*, June 2012, pp. 796–801.

[10] S. Venkataramani, K. Roy, and A. Raghunathan, "Substitute-and-simplify: A unified design paradigm for approximate and quality configurable circuits," in *2013 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2013, pp. 1367–1372.

[11] J. Miao, A. Gerstlauer, and M. Orshansky, "Multi-level approximate logic synthesis under general error constraints," in *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov 2014, pp. 504–510.

[12] Y. Wu and W. Qian, "An efficient method for multi-level approximate logic synthesis under error rate constraint," in *2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, June 2016, pp. 1–6.

[13] D. Shin and S. K. Gupta, "A new circuit simplification method for error tolerant applications," in *2011 Design, Automation Test in Europe*, March 2011, pp. 1–6.

[14] A. Ranjan, A. Raha, S. Venkataramani, K. Roy, and A. Raghunathan, "Aslan: Synthesis of approximate sequential circuits," in *2014 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2014.

[15] L. Holik, O. Lengal, A. Rogalewicz, L. Sekanina, Z. Vasicek, and T. Vojnar, "Towards formal relaxed equivalence checking in approximate computing methodology," *2nd Workshop On Approximate Computing (WAPCO)*, 2016. [Online]. Available: https://wapco.ece.uth.gr/2016/papers/SESSION2/wapco2016_2_1.pdf

[16] A. Chandrasekharan, S. Eggersglüß, D. Große, and R. Drechsler, "Approximation-aware testing for approximate circuits," in *2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan 2018, pp. 239–244.

[17] M. Traiola, A. Virazel, P. Girard, M. Barbareschi, and A. Bosio, "Testing approximate digital circuits: Challenges and opportunities," in *2018 19th IEEE Latin American Test Symposium (LATS)*, March 2018.

[18] V. Mrazek, R. Hrbacek, Z. Vasicek, and L. Sekanina, "Evoapprox8b: Library of approx adders and multipliers for circuit design and benchmarking of approximation methods," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, March 2017, pp. 258–261.