Increasing Circuit Lifetime by Accepting Precise Adders to work as Approximate Ones

W. J. Perez-Holguin
UPTC
GIRA Group
Sogamoso, Colombia
wilson.perez@uptc.edu.co

E. Sanchez
Politecnico di Torino
DAUIN
Torino, Italy
ernesto.sanchez@polito.it

***Abstract*— Approximate computing is today one of the hottest topics related to circuit design and optimization, since by adopting this approach, designers are able to reduce area, power consumption, and even production costs in the case the target application is able to accept a given degree of inaccuracy in the final computations. In this paper, on the contrary, we propose a novel idea intended to analyze how accurate circuits behave in presence of faults and understanding if it is possible to still accept the faulty device in the given application, increasing, in this way, the device lifetime. We gather a set of results resorting to a set of 8-bit adders widely used in commercial applications in order to experimentally demonstrate the method suitability.**

Keywords*— *Approximate Computing, Approxiamte Testing, 8-bit precise adders, reliability.

I. INTRODUCTION

In recent years, the Approximate Computing (AC) has emerged as a new paradigm aimed to assess the use of systems that perform inaccurate calculations rather than systems that generate guaranteed accurate results. Notably, for many applications, this type of system does not produce a significant drop in its overall performance, producing instead benefits such as a decrease of production costs, higher yield, and fewer energy requirements [1][2][3].

Working on the AC approach represents the need for the development of modified versions of the hardware components like memories, adders, multipliers, etc., and carefully adapted software layers, which normally require fewer gates counting and code lines, leading to a reduction of energy and memory space requirements, reason why approximate computing has become, not merely attractive but even imperative [1][4].

Some of the basic techniques employed for digital circuits design comprise the *over-scaling* based approximation and the *functional approximation*. In the first one, the IC works out of its nominal operating conditions, reducing, for example, the voltage supply to a minimum value. This generates a substantial energy saving but can produce timing errors at the output [1]. On the other hand, functional approximation aims at modifying the circuit structure. This leads to an area and energy saving, but at the same time it reduces the accuracy and produces errors in the output. Manual and automatic approaches have generally been used for the functional approach, including evolutionary algorithms for the generation of the last ones [5][6][7][8]. In this scenario, it is important to clearly define the metric used to assess the obtained results. Even though error metrics can be problem dependent, some different techniques have been used resorting to *Error Probability*, *Average Error Magnitude*, *Maximal Relative Error*, and *Error Magnitude* or *Worst Case Error*.

Regarding digital circuits testing, few approaches have been proposed oriented to qualify the test generation patterns created through approximate testing. Interestingly, in [10] the authors propose a technique able to automatically generate a set of test patterns guaranteeing that the corresponding circuit complies whit a given maximum error. The proposed approach was evaluated in a set of approximate adders and multipliers freely available in [9] showing the capacity of the proposed technique to substantially reduce the number of test patterns according the accepted error.

In this paper, we propose for the very first time the possibility to use accurate circuits as approximate ones in the case these are affected by one fault. The adoption of the proposed idea requires to perform an initial analysis on the accurate circuits in order to understand how to determine the actual circuit accuracy degradation in presence of a fault. In addition, assuming that the accurate circuit is already in-filed, it is necessary to define how to measure the device degradation and then deciding if this is still acceptable to use the circuit or not.

The rest of the paper is organized as follows: Section II describes the proposed approach and provides a basic example. Section III analyzes a set of 7 8-bit adders considering the proposed approach. And finally, the conclusions are given in Section IV.

II. PROPOSED APPROACH

The main goal of this paper is to propose for the very first time the possibility to allow faulty devices to work as approximate ones, intended to enlarge the operational life of the device by accepting the possibility to decrease accuracy.

In order to achieve the proposed goal, it is necessary to accomplish with two different steps:

1. Accurate analysis towards approximate
2. In-filed diagnostic testing.

The first step is performed offline and it consists on a preliminary analysis of the accurate circuit in presence of faults. This step lets us with a clear view of how the device behaves in presence of a given fault, indicating the worst error that may be produced by the faulty circuit with respect to the correct circuit.

The second step is performed in-field and requires to use a set of test patterns able to identify the degradation degree suffered by the circuit do to the fault. Then, in case the circuit degradation is lower than the maximum admissible one, it is possible to use the circuit as an approximate one; on the contrary case, the circuit degradation is so high and is not possible to further accept the circuit results, and the device must be substituted.

In the following, an example developed in a very simple 2-bit precise adder is presented in order to clarify the main idea proposed here.

A complete analysis in the 2-bit adder is shown in Figure 1. The basic idea is to analyze the adder behavior when affected by a given fault, checking for any input, the circuit outputs and comparing the results against the expected ones. In the proposed approach, the stuck-at faults were only studied.

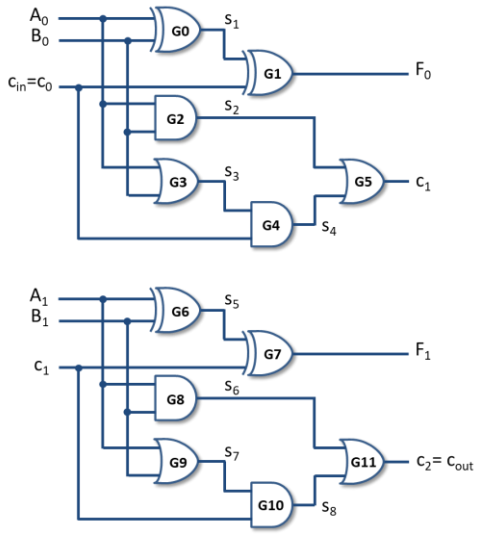

Figure 1. 2-bit precise adder

For example, Table I shows the circuit behavior considering a couple of faults, firstly, the stuck-at 0 in the upper input of gate G1 (columns 3 and 4), and the stuck-at 1 in the upper input of gate G10 (columns 5 and 6). The table first column reports all the possible input values in input signals *A* and *B*, as well as *Cin*. The second column indicates the expected value (*Golden*) whereas the third and fifth columns report the faulty circuit outputs, and, columns fourth and sixth report the absolute value of the resulting error.

From the table, it is possible to notice that checking the 32 input vectors, in the case of the S@0 in the upper input of G1, the faulty adder outputs are correct 10 times out of 32. 10 times the absolute error equals 1, 4 times the error is 3, 6 times the error equals 4, and the last 2 times the error equals 5. Then, assuming that we may accept one fault, the circuit behavior is correct 31.25% of the times and the worst case error is 5.

In the case of the second fault, the S@1 in the upper input of G10, it is interesting to notice that the circuit outputs are correct 25 times, and the absolute error is 4 for the remaining 7 input vectors. This lets us with a faulty circuit answering correctly 78.12% of the times, and having a worst case error equal to 4.

TABLE I. CIRCUIT OUTPUTS ON 2 DIFFERENT FAULTS

A B Cin	Golden	Faulty G1 S@0inX		Faulty G10 S@1inX	
		Output	ABS Error	Output	ABS Error
00|00|0	000	000	0	000	0
00|00|1	001	001	0	001	0
00|01|0	001	000	1	001	0
00|01|1	010	011	1	110	4
00|10|0	010	010	0	010	0
00|10|1	011	011	0	111	4
00|11|0	011	010	1	111	4
00|11|1	100	001	3	100	0
01|00|0	001	000	1	001	0
01|00|1	010	011	1	110	4
01|01|0	010	010	0	110	4
01|01|1	011	011	0	111	4
01|10|0	011	010	1	111	4
01|10|1	100	001	3	100	0
01|11|0	100	000	4	100	0
01|11|1	101	001	4	101	0
10|00|0	010	010	0	010	0
10|00|1	011	011	0	011	0
10|01|0	011	010	1	011	0
10|01|1	100	001	3	100	0
10|10|0	100	100	0	100	0
10|10|1	101	001	4	101	0
10|11|0	101	000	5	101	0
10|11|1	110	111	1	110	0
11|00|0	011	010	1	011	0
11|00|1	100	001	3	100	0
11|01|0	100	000	4	100	0
11|01|1	101	001	4	101	0
11|10|0	101	000	5	101	0
11|10|1	110	111	1	110	0
11|11|0	110	110	0	110	0
11|11|1	111	011	4	111	0

As for the previous two faults, a complete analysis were done in the 2-bit adder exhaustively, including all the possible stuck-at faults and checking the circuit outputs for any input vector. Figure 2 reports the percentage values of the *Absolute Error* obtained by computing the output of the faulty circuit against the expected one. Interestingly, in about 52% of the cases, the circuit answers are the correct ones, while only in about 1% of the times the circuit reaches its maximum error in the outputs.

Once identified the circuit behavior in presence of an error, we need to create the in-field test patterns able to diagnose the circuit providing the degradation level due to a possible fault. this step can be performed by using a technique similar to the one presented on [10] and [11]; however, some additional modifications are required since the proposed method is not suitable to find the suitable test patterns for in-field testing but post production testing.

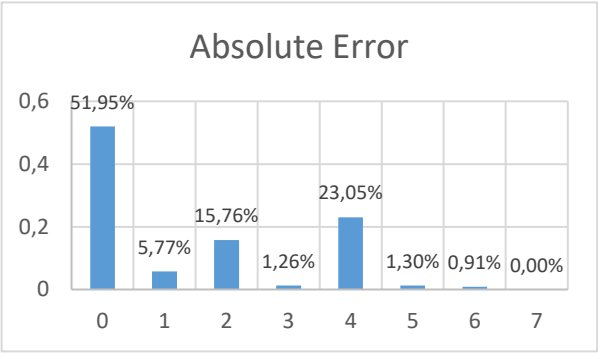

Figure 2. 2-bit Adder Output Absolute Error

Therefore, assuming that in our application we may accept the use of approximate adders having a worst case error equal to 4, the faulty adder analyzed before could still be a feasible solution.

Clearly, in order to approach the circuit analysis towards approximate an exhaustive approach is not affordable for larger circuits. Hence, it is important to find a solution that allows us to understand which faults may leave us with a precise circuit behaving as an approximate one.

A. *Proposed framework*

In order to analyze the possibility of using a faulty circuit as an approximate one, we propose the framework depicted in Figure 3, the proposal is similar to the one reported in [6], however, in that work, the idea is to check an approximate precise adder against a precise one.

In the proposed framework, the circuit inputs (X_0, X_1,… X_n) are provided to a couple of precise adders, of which the second is the faulty one. The circuits' outputs *Y* for the precise adder, and *Y'* for the faulty one, are compared and the absolute error *E* is computed. Then, the obtained value is compared against a given error ε that represents the Worst Case Error allowed in the actual circuit. Then, the output answer is OK in the case the faulty circuit produces an output which error is lower than the accepted one.

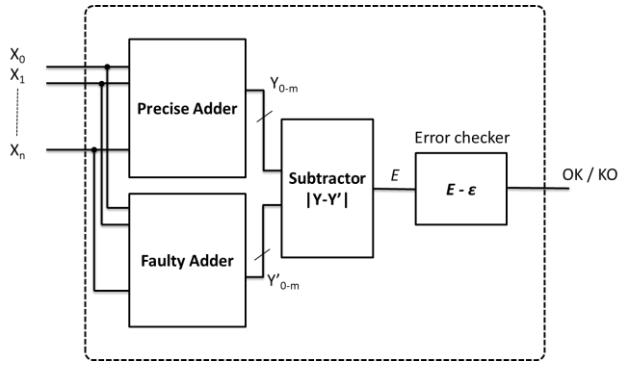

Figure 3. Proposed framework

III. CASE STUDY

Resorting to the proposed approach, we analyze the set of 7 widely used 8-bit adders. In the reported experiments, the adders were synthesized using a 180 nm technology library. The next table reports the area occupation in μm^2, the estimated power in µW, and the number of faults considering only the stuck-at ones.

The preliminary results provided here, only include the analysis step described before not including the diagnostic part for in-filed testing.

TABLE II. 8-BIT ADDERS CHARACTERISTICS

8-bit adders	AREA [um^2]	POWER [uW]	Num faults
Carry Select (CarrySel)	1608	641.90	762
Ripple Carry (RippCarry)	920	339.90	462
Carry Lookahead (CarryLKH)	3080	1004.30	762
Higher Valency Tree Adder with HanCarlson Architecture (HVTrHCA)	1856	736.80	534
Higher Valency Tree Adder with Kogge-Stone Architecture (HVTrKSA)	2368	912.20	630
Tree Adder with HanCarlson Architecture (TwHCA)	1952	751.50	552
Tree Adder with Kogge-Stone Architecture (TwKSA)	2240	841.10	606

In order to evaluate the different circuits' behavior when affected by a fault, we implemented the proposed approach resorting to a commercial fault simulator. The proposed framework could be implemented also by using a logic simulator; however, the simulation time becomes prohibitive. Thus, we decide to implement the proposed approach exploiting a fault simulator, injecting the faults in the second instance of the precise adder only.

For any precise adder reported in the previous table, we synthesize a circuit as the one proposed in Figure 3. *Proposed framework*, changing at every experiment the error value ε. In our experiments ε ranges from 1 to 260. Resorting to the proposed scheme, it is possible to say that a faulty circuit that produces an output error lower than the given threshold ε will produce the correct output (*OK*), meaning that the fault was not covered. On the contrary, if the faulty adder error is higher than the expected one, the circuit output is wrong (*KO*) detecting in this case the injected fault.

For any circuit we ask the fault simulator tool to generate the set of combinational test patterns able to cover the adder faults. This test set of testing patterns is suitable for post production, not for in-field testing; for this reason, we do not report information about the test patterns in the following. After any run, we only collected the number of covered faults for any given error value and report in the following figures the obtained results.

Interestingly, it is possible to note that for all the analyzed adders, almost the 90% of faults generate output errors higher than 2. On the other side, it is also possible to note that very few faults produce error higher than 150.

Figures 4 and 5 report the fault analysis result obtained on our experiments. The figures report in the X axis the threshold error ε; while, in the Y axis reports the percentage of faults that generate an output with an absolute error higher than ε.

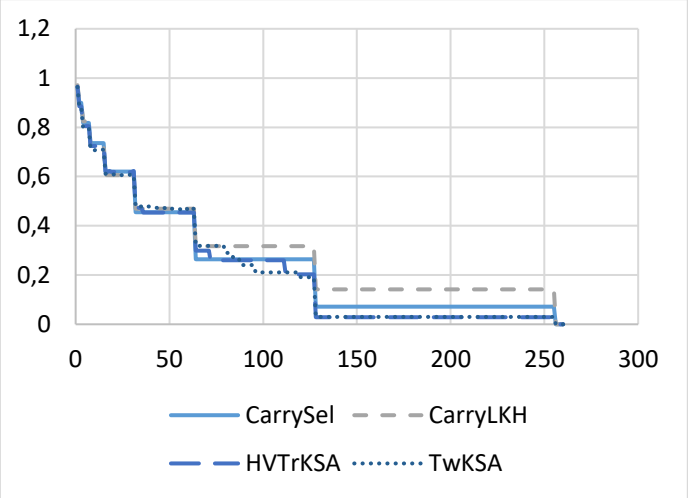

Figure 4. Faults analysis for CarrySel, CarryLKH, HVtrKSA and TwKSA 8-bit adders

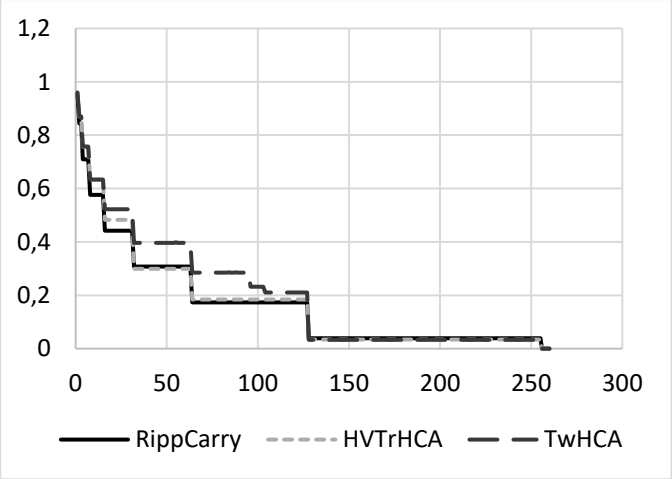

Figure 5. Fault analysis for Ripple Carry, HVTrHCA, and TwHCA 8-bit adders.

In particular, considering the Higher Valency Tree Adder with HanCarlson Architecture (HVTrHCA) and the Ripple Carry adder (see figure 5), in the case a WCE of 50 is accepted, only 30% of the faults will generate output errors higher than 50.

IV. Conclusions

In this paper, we propose a novel technique to improve the circuit reliability by accepting during the operational time to use the precise circuit as an approximate one. The main assumption done here is that the faulty circuit under a certain conditions will be able to operate as an accurate one even if affected by a set of faults.

The proposed methodology was preliminary evaluated on 7 8-bit precise adders showing the possibility to accept a circuit degradation that mainly depend on the actual application.

Interestingly, it is possible to observe that in the case the elaboration accuracy may accept a Worst Case Error up to 50, the some 8-bit adders may be able to accept up to 70% of the circuit faults.

We are currently extending the proposed approach to analyze a set of 8-bit multipliers, and a set of approximate adders and multipliers in order to improve the comparison of our current experiments.

REFERENCES

[1] S. Mittal, “A survey of techniques for approximate computing,” *ACM Comput. Surv.*, vol. 48, no. 4, p. 62, 2016.

[2] Q. Xu, T. Mytkowicz, and N. S. Kim, “Approximate computing: A survey,” *IEEE Des. Test*, 2018.

[3] J. Han and M. Orshansky, “Approximate computing: An emerging paradigm for energy-efficient design,” in *Test Symposium (ETS), 2013 18th IEEE European*, 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*, 2013, p. 113.

[5] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, “Design and analysis of approximate compressors for multiplication,” *IEEE Trans. Comput.*, vol. 64, no. 4, pp. 984–994, 2015.

[6] P. Kulkarni, P. Gupta, and M. Ercegovac, “Trading accuracy for power with an underdesigned multiplier architecture,” in *VLSI Design (VLSI Design), 2011 24th International Conference on*, 2011, pp. 346–351.

[7] D. Shin and S. K. Gupta, “A new circuit simplification method for error tolerant applications,” in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2011*, 2011, pp. 1–6.

[8] D. Shin and S. K. Gupta, “Approximate logic synthesis for error tolerant applications,” in *Proceedings of the Conference on Design, Automation and Test in Europe*, 2010, pp. 957–960.

[9] V. Mrazek, R. Hrbacek, Z. Vasicek, and L. Sekanina, “Evoapproxsb: 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.

[10] Marcello Traiola, Arnaud Virazel, Patrick Girard, Mario Barbareschi, Alberto Bosio, "Testing Approximate Digital Circuits: Challenges and Opportunities", in Latin American Test Symposium 2018, pp.

[11] 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.