Input-aware accuracy characterization for approximate circuits

Ali Piri, Salvatore Pappalardo, Salvatore Barone, Mario Barbareschi, Bastien Deveautour, Marcello Traiola, Ian O'Connor, Alberto Bosio

Input-aware accuracy characterization for approximate circuits

Ali Piri[1], Salvatore Pappalardo[1], Salvatore Barone[2], Mario Barbareschi[2], Bastien Deveautour[1], Marcello Traiola[3], Ian O'Connor[1], Alberto Bosio[1]
[1]*Univ Lyon, ECL, INSA Lyon, CNRS, UCBL, CPE Lyon, INL, UMR5270, 69130 Ecully, France*
[2]*Dep. of Electrical Engineering and Information Technology, University of Naples Federico II, Naples, Italy*
[3]*University of Rennes, Inria, CNRS, IRISA, UMR6074*
Email: [1]{name.surname}@ec-lyon.fr, [2]{name.surname}@unina.it, [3]marcello.traiola@inria.fr

***Abstract*—It has been a while since Approximate Computing (AxC) is applied systematically at various abstraction levels to increase the efficiency of several applications such as image processing and machine learning. Despite its benefit, AxC is still agnostic concerning the specific workload (i.e., input data to be processed) of a given application. For instance, in signal processing applications (such as a filter), some inputs are constants (filter coefficients). Meaning that a further level of approximation can be introduced by considering the specific input distribution. This approach has been referred to as "input-aware approximation". In this paper, we explore how the input-aware approximate design approach can become part of a systematic, generic, and automatic design flow by knowing the data distribution. In particular, we show how input distribution can affect the error characteristics of an approximate arithmetic circuit and also the advantage of considering the data distribution by designing an input-aware approximate multiplier specifically intended for a high-pass FIR filter, where the coefficients are constant. Experimental results show that we can significantly reduce power consumption while keeping an error rate lower than state-of-the-art approximate multipliers.**

***Index Terms*—Approximate Computing, Energy Efficiency, Embedded Systems, Input-Aware Approximation**

I. Introduction

The performance and efficiency of a number of time- and resource-intensive applications can be significantly reduced by using the AxC design paradigm, as shown in the scientific literature [1]. AxC exploits various sources of error resilience in applications by systematically trading the quality of output results for performance gains or resource savings [2]. AxC has been successfully deployed in several domains, for both software and hardware applications, including signal and image processing, Big Data analytics, and machine learning. To realize the full potential of AxC, many challenges must be overcome, all of which are a direct result of the need to consider the target application. These include (i) identifying parts of the application that are suitable for approximation, and (ii) appropriate approximation techniques, then (iii) choosing an appropriate error metric for error evaluation, and finally (iv) selecting those approximation configurations that provide an optimal balance between the introduced error and the resulting benefit.

As for the arithmetic operators, a large number of scientific works resort to manual or automatic methods to design approximate components, mainly aiming at saving hardware resources. Although these operators have been successfully used in numerous applications, including image processing [3], [4], machine learning, and artificial intelligence [5], [6], adapting the components to their final application opens up further possibilities for approximation, as shown in recent papers. In fact, additional efficiency cannot be achieved if generalization and multiple use are the goals of approximation flows. Conversely, exploiting the typical workload of a given application by a finer-grained approximation of its components can enable better results in terms of both errors and performance.

In this paper, we show that it is possible to take advantage of the data distribution to obtain a fine-tuned approximation and thus achieve better results (i.e., higher energy efficiency, lower latency and area overheads) with a lower impact on the application accuracy. In particular, by characterizing the input distribution pattern, we can design an approximate version of a given circuit by exploiting the fact that some inputs with similar characteristics are elaborated more frequently than others. Therefore, in this paper we present an input-aware approximate multiplier specifically designed for a case study (high-pass FIR filter) where the input distribution pattern is not uniform. We also discussed how the accuracy of the multiplier operating in such an environment should be characterized.

The remainder of this paper is organized as follows: In section II, we first present a case study with a non-uniform input distribution and the input-aware approximate multiplier developed for this application. We then discuss how to characterize the accuracy in the presence of non-uniform data distribution. In section III, the experimental setup for the evaluation and comparison is presented and the results are discussed. Finally, conclusions are drawn in section IV.

II. input-aware approximation

The application that we have targeted is a Finite-Impulse Response (FIR) filter (1), where x_j is the j^{th} sample of the input signal, b_i is the i^{th} coefficient of the signal and y_n is the n^{th} sample of the output signal. FIRs are one of the most common signal processing applications where data distribution pattern might not be uniform.

$$y_n = \sum_{i=0}^{N} b_i \cdot x_{n-i} \tag{1}$$

The goal of this case study is to design an 8-bit multiplier that performs the $b_i \cdot x_{n-i}$ part of the equation (1) while minimizing both the approximation error and the hardware requirements. The FIR filter considered is a low-pass filter with a cutoff frequency of 500 Hz and an attenuation of 60 dB in the stopband. This FIR filter uses the Q1.7 signed fixed-point arithmetic format where there is one bit for the integer part and seven bits for the fractional part. The output of this multiplication is in Q2.14 signed fixed-point format. There are only five coefficients for this filter with different probabilities of occurrence over the other coefficients. The coefficients and their probabilities are shown in Table I.

TABLE I: Coefficients of the FIR filter (in binary) and their probability

Coefficients (Binary)	Coefficients (Decimal)	Probability
00000000	0	0.0027
00000011	0.0234375	0.0523
00001110	0.109375	0.2157
00011101	0.2265625	0.4478
00100100	0.28125	0.2815

A. Input-aware approximate multiplier design

To implement a signed multiplier working with this FIR filter, we adopted the Baugh-Wooley algorithm and the input-aware approximate method used in [7]. As can be seen in Table I, for all coefficients of this particular FIR filter, the two most significant bits are always logic-0. Therefore, it is not necessary to calculate the relative partial products. The simplest way to design an input-aware multiplier for this application is to cut out the two corresponding columns of AND gates in the partial product tree of the multiplier, assuming that the other multiplicand is uniformly distributed. To further approximate the multiplier, we can also eliminate the third column since it is logic-1 in only one of the coefficients. However, given the weight of the bit and the probability of the input occurring, this could severely affect the accuracy.

B. Error metrics in Input-Aware approximate circuits

In the state of the art, since the presented approximate arithmetic circuits were designed for workloads where the input distribution was considered uniform, they have usually calculated the error metrics with random or all possible inputs [8], [9]. As discussed before, In the applications that we are aiming for, the input distribution is not uniform. Meaning that some inputs are more probable to happen and some may not happen at all. So, for calculating the error metrics we need to take the probability of the input scenarios into account.

Error Distance (ED) and Error Probability (EP) are the basic error metrics in approximate computing. ED is the distance between the correct output and the approximate output of a circuit for each scenario and EP is the probability of having a wrong answer which is calculated by number of wrong answers over number of all input scenarios. The Mean Error Distance (MED) is the average of all the error distances which is calculated as (2) where N is the number of possible scenarios, $O_{Acc}^{(i)}$ and $O_{AxC}^{(i)}$ are the accurate and approximate output of i^{th} scenario respectively, and $P(S_i)$ is the probability of the i^{th} scenario happening. For instance, [10] has presented an approximate 2-bit multiplier where error only happens when both the inputs are binary values of "11" (3 in decimal) which the result is supposed to be 9. So instead of "1001" the output is "111" (7 in decimal). In this case, the error distance is 2 and assuming that inputs are uniformly distributed, the probability of this scenario to happen is 1/16. We also have Mean relative Error Distance (MRED) which is the average of error distances relative to the correct answers (3). And then we have Absolute Worst-Case Error (AWCE) which is the largest error distance that can happen (4).

$$MED = \sum_{i=1}^{N} \frac{|O_{AxC}^{(i)} - O_{Acc}^{(i)}|}{N} \cdot P(S_i) \tag{2}$$

$$MRED = \sum_{i=1}^{N} \frac{|O_{AxC}^{(i)} - O_{Acc}^{(i)}|}{|O_{Acc}^{(i)}|} \cdot P(S_i) \tag{3}$$

$$AWCE = \max_{\forall i} |O_{AxC}^{(i)} - O_{Acc}^{(i)}| \tag{4}$$

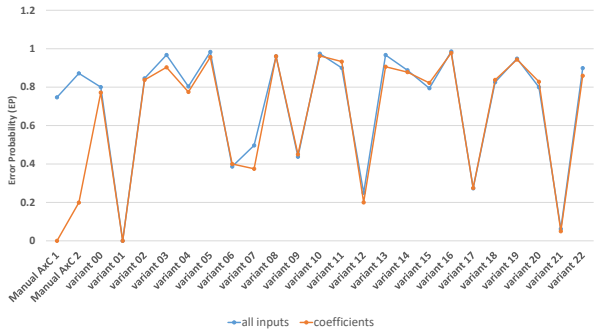
In Section III, we will first show how the distribution pattern of the inputs and the probability of them can affect the accuracy metrics of an approximate multiplier. Then we will compare our input-aware multiplier to a set of non-input-aware approximate multipliers to see how they behave in terms of accuracy and circuit properties, and how they affect the FIR filter.

III. Evaluation

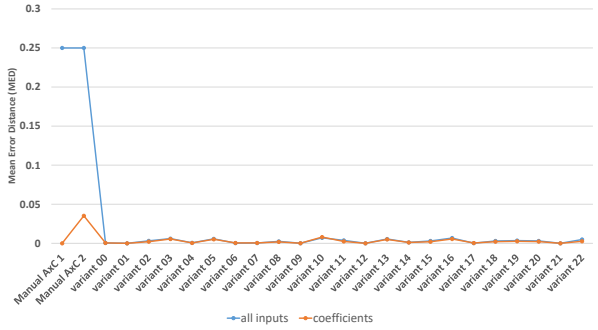
To demonstrate the efficiency of our input-aware multiplier we have compared it with some non-input-aware multipliers generated with the tool from [3] which exploits the And-Inverter Graph (AIG) representation of digital circuits in order to introduce approximation. We have generated twenty four non-input-aware multipliers with varying accuracies using this tool.

A. Accuracy characterization

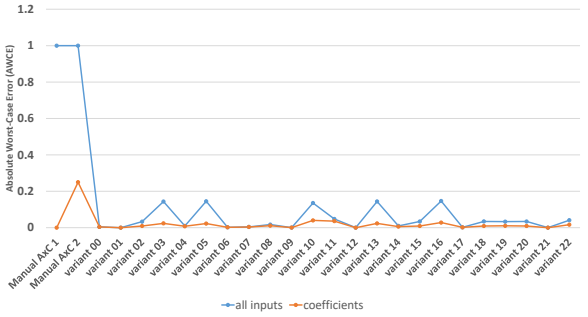
To show how the distribution of inputs can affect the error metrics of an approximate multiplier, we simulated the automatically generated multipliers and the input-aware multipliers, once with all possible input scenarios for an 8-bit multiplier and once with the coefficients only. The multipliers were generated in verilog and were simulated using verilator [11]. We calculated the EP, MED and, AWCE using the formulas described in Section II-B. "Manual AxC 1" is the input-aware multiplier where we have only eliminated the part of the multiplier which won't affect the accuracy when working with the given workload and "Manual AxC 2" is the one where the third column of and gates is also cut out of the



(a) Error Probability (EP)



(b) Mean Error Distance (MED)



(c) Absolute Worst-Case Error (AWCE)

Fig. 1: Input-aware and Non-input-aware accuracy characterization of the multipliers.

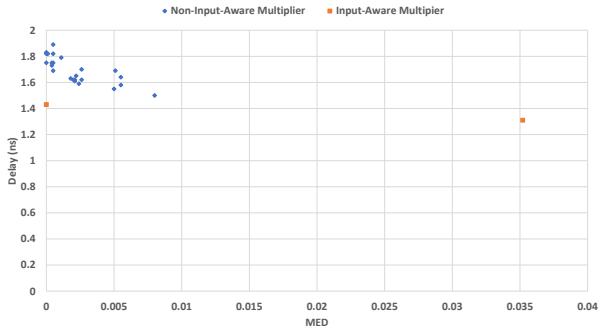
design and will lead to an error when the third most significant bit of the coefficient is logic-1.

As it can be seen in Figure 1 EP is zero with our input-aware multiplier. even for the non-input-aware approximate multipliers, EP is less when we only calculate it for the given workload. For the MED, it is the same case, except for the input-aware multipliers where error distances are very high when MED is calculated using all the possible scenarios for an 8-bit multiplier because the error happens in the most significant bits. And finally, its obvious that AWCE, when only the coefficients considered, will always be lower than or equal with the AWCE of all scenarios.

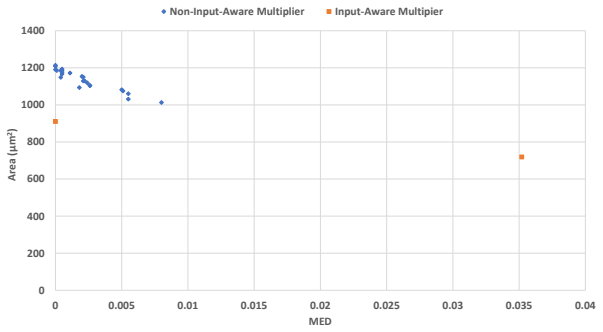
B. Circuit characterization

For the circuit characterization we have calculated the Power consumption, delay, and area of these 8-bit multipliers by synthesizing them using the Synopsys Design Compiler (DC) at 45nm. The designs were implemented in Verilog at the gate level. These results are presented in Figure 2 where MED used for comparison here is calculated for the given coefficients.

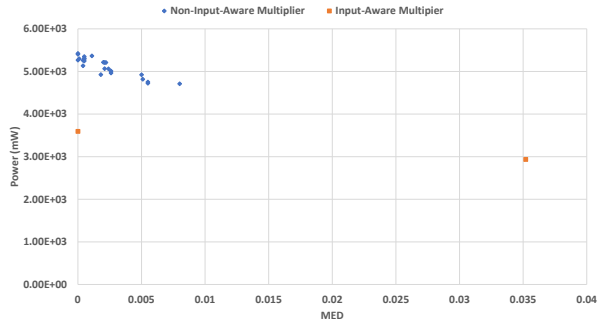
For the automatically generated non-input-aware multipliers the more MED gets, delay, power consumption, and area reduces. On the other hand, For the first Input-aware multiplier (Manual AxC 1) MED is equal to zero since for the given workload it is accurate and yet, it requires 20% less silicon area and consumes 30% less power compared to the non-input-aware approximate multipliers. however, for the approximate version (Manual AxC 2) MED is very high since approximation is done on a significant bit.



(a) Delay-MED



(b) Area-MED



(c) Power-MED

Fig. 2: Circuit characteristics of the multipliers to their MEDs for the given coefficients.

In order to evaluate our approach on application level, we

assessed the impact of approximation on the FIR filter by measuring the Peak Signal-to-Noise Ratio (PSNR) between the output signals produced by the accurate and approximate filters. Furthermore, we compare FIR filter implementations using multipliers generated automatically from the approach presented in [3] against our input-aware multipliers. Such a comparison considers the error measured using the PSNR and the actual hardware requirements while targeting the 45nm standard cell library [12].

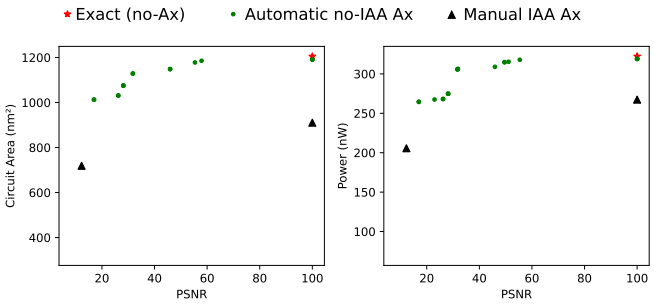


Fig. 3: PSNR and hardware resources for 8-bits FIR while using approximate multipliers

As shown in Figure 3 which report results for the 8-bits FIRs, Green dots denote multipliers resulting from the automatically generated with non-input aware method, while the black triangles denote our input-aware multipliers and the red star denotes the original, accurate multiplier. As it can be observed, the results from our input-aware approach performs better since even the accurate input-aware multiplier requires less silicon area and consume less power compared to the approximate multiplier generated automatically.

IV. Conclusion

In this paper, we discussed how knowing the workload of an application can help input-awareness become part of the approximate circuit design flow which will lead to more efficient computing systems. In particular, we presented an input-aware multiplier specially designed for a FIR filter with non-uniform input distribution. We discussed how accuracy should be characterized in such input-aware approximate circuits and we compared the efficiency of our design with non-input-aware approximate multipliers and proved that input-aware circuits resulting from our method perform better than those resulting from non-input-aware methodologies in terms of accuracy, power consumption, circuit area, and performance. The FIR filter is only one of several applications that have non-uniform distribution for their inputs. There are more applications with more possibilities for approximation. The input-aware method can also be included in the automatic approximate design flow.

Acknowledgment

This work has received funding from the APROPOS project in the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 956090.

References

[1] A. Bosio, D. Ménard, and O. Sentieys, Eds., *Approximate Computing Techniques: From Component- to Application-Level*. Cham: Springer International Publishing, 2022. [Online]. Available: https://link.springer.com/10.1007/978-3-030-94705-7

[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] M. Barbareschi, S. Barone, N. Mazzocca, and A. Moriconi, "A Catalog-based AIG-Rewriting Approach to the Design of Approximate Components," *IEEE Transactions on Emerging Topics in Computing*, 2022.

[4] M. Barbareschi, S. Barone, A. Bosio, J. Han, and M. Traiola, "A Genetic-algorithm-based Approach to the Design of DCT Hardware Accelerators," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 18, no. 3, pp. 1–25, Jul. 2022.

[5] V. Mrazek, M. A. Hanif, Z. Vasicek, L. Sekanina, and M. Shafique, "autoAx: An Automatic Design Space Exploration and Circuit Building Methodology utilizing Libraries of Approximate Components," in *2019 56th ACM/IEEE Design Automation Conference (DAC)*, Jun. 2019, pp. 1–6, iSSN: 0738-100X.

[6] V. Mrazek, L. Sekanina, and Z. Vasicek, "Libraries of Approximate Circuits: Automated Design and Application in CNN Accelerators," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, no. 4, pp. 406–418, Dec. 2020.

[7] A. Piri, S. Saeedi, M. Barbareschi, B. Deveautour, S. D. Carlo, I. O'Connor, A. Savino, M. Traiola, and A. Bosio, "Input-Aware Approximate Computing," in *2022 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, May 2022, pp. 1–6.

[8] H. Jiang, F. J. H. Santiago, H. Mo, L. Liu, and J. Han, "Approximate arithmetic circuits: A survey, characterization, and recent applications," *Proceedings of the IEEE*, vol. 108, no. 12, pp. 2108–2135, 2020.

[9] H. Jiang, C. Liu, L. Liu, F. Lombardi, and J. Han, "A review, classification, and comparative evaluation of approximate arithmetic circuits," *J. Emerg. Technol. Comput. Syst.*, vol. 13, no. 4, aug 2017. [Online]. Available: https://doi.org/10.1145/3094124

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

[11] W. Snyder, "Verilator and systemperl," in *North American SystemC Users' Group, Design Automation Conference*, 2004.

[12] "FreePDK45 | NC State EDA." [Online]. Available: https://eda.ncsu.edu/freepdk/freepdk45/