Exploiting Parallel Prefix Adders in the Approximate Adders for Low-Power SAD Metric

Morgana Macedo , Bianca Silveira , Leonardo Soares, *Member, IEEE,* Cláudio Diniz, *Member, IEEE,* and Eduardo Costa, *Member, IEEE.*

***Abstract*—The approximated adders are generally composed of two parts, that is, the imprecise (approximate) part, which is a set of least significant bits, and a precise part, with the remaining most significant bits. Previous work has only explored the Ripple Carry Adder (RCA) in the precise part of approximate adders. This work proposes the exploration of different topologies of parallel prefix adders (PPA) in the precise part of approximate adders, whose approximate adders are introduced in the sum tree of the Sum of Absolute Differences (SAD) metric, aiming at the ideal combination of performance and dissipation power. In addition to the RCA, we evaluated the use of PPAs such as Brent-Kung, Kogge-Stone, Han Carlson, Ladner-Fisher and Sklansky in the precise part of two well-known approximate adductors, that is, Error Tolerant I (ETA-I) and copy adder. The additives were described in VHDL and synthesized in 45 nm CMOS technology. Synthesis results point to the Ladner Fischer adder as the most appropriate to be used in the precise part of the approximated adders.**

Keywords*—*Approximate Computing, Parallel Prefix Adders, Approximate Adders, Low Power, SAD, VLSI design.

I. Introduction

Motion Estimation (ME) is one of the main responsible for the high compression ratio in digital videos, which is why it is one of the most important steps in coding. The ME aims to eliminate temporal redundancy. Given a block of the frame to be coded and a search window in a frame of reference, the ME looks for the block of the search window that is more similar to the block of the frame to be coded.

During ME a similarity metric is used to find the best match between the blocks being coded and the candidate blocks, operation is known as Block Matching (BM). Among the existing similarity metrics, the sum of the absolute differences (SAD) (figure 1) used to implement hardware encoders.

In this work we will explore the sum tree of the SAD (figure 2), using the approximated adders, which are exploited by efficient adders PPA (Parallel Prefix Adder) in its precise part. Among the PPA adders explored we mention Brent Kung [1], Kogge Stone [2], Han Carlson [3], Sklansky [4] and Ladner Fischer [5]. SAD architectures developed with 16, 24 and 32-bit widths, use in their sum tree approximate sums of 16, 24 and 32 bits, respectively. The approximated 16, 24, and 32-bit modulators are divided into two parts the precise and the approximate part, both the precise and the approximate part have the same width, in half the bits, for example, in a 16-bit bits, 8 bits are accurate and 8 bits are approximate.

The approximate computing [6] has emerged as an interesting area for energy efficiency, since computationally intensive

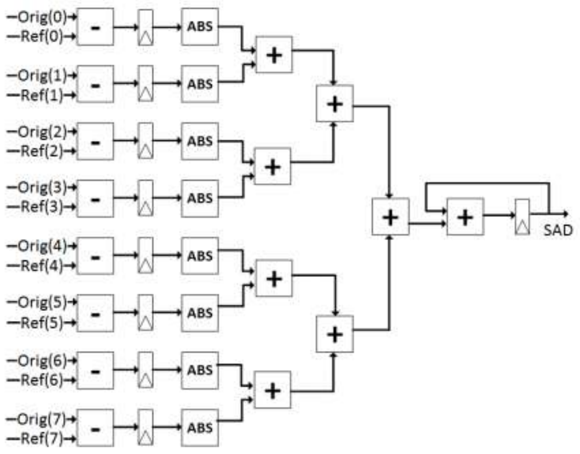

Fig. 1. Block diagram of the SAD architecture using the ”sum” tool.

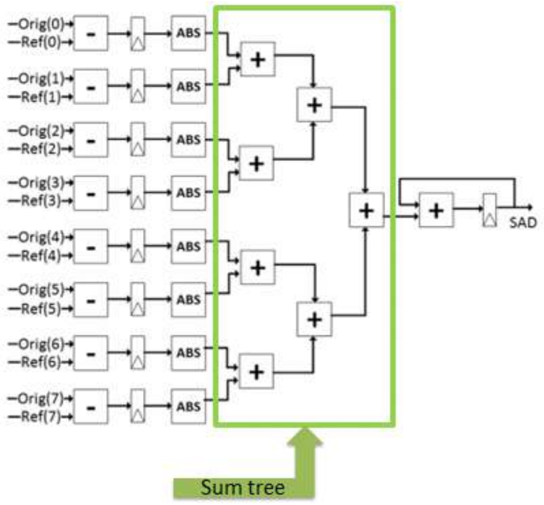

Fig. 2. Sum tree explored with the use of approximate adders.

M. Macedo, B. Silveira, C. Diniz and E.Costa was with the Graduate Program on Electronic Engineering and Computing - Catholic University of Pelotas (UCPel), Pelotas, Brazil. e-mail: (morganamacedoazevedodarosa@gmail.com).

L. Soares was with the Graduate Program on Microelectronics (PGMicro) - Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Brazil.

Manuscript received April 19, 2005; revised August 26, 2015.

applications like visual processing and multimedia signals do not require high precision to work correctly. Adders circuits have been explored in approximate computing context since they are basic building blocks heavily used in those intensive computational applications. They can be implemented in a way to balance the trade-off between accuracy vs. performance/power.

Approximate computing is a solution for energy-saving projects, providing balance between accuracy and power [7]. Therefore, both the error distance and the error rate are important for the quality of the output. In the approximate computation the approximate summers are divided in two groups: (I) exact part located in the most significant bits; (II) approximate part located at the least significant bits. For each less significant bit there are techniques that implement an inexact addition. What is common, according to [8], in the inexact part of addition that simplification has a design of a full adder. In the most significant bits we will explore for better performance the PPA (Parallel Prefix Adder).

The approximate computing paradigm emerged to increase performance and to reduce power dissipation [6]. The key approach in approximate hardware is to reduce the computation accuracy in favor of energy-efficiency. In circuit level design, this is performed by designing simpler circuits to speed up the critical path timing and/or to consume less power. Approximate computing techniques take advantage of approximation-tolerant applications which do not need high accuracy all the time but only “good enough” or “sufficiently good” results for output perceptual quality. In [9] is stated the following properties to define an approximation-resilient application: (i) there is no a golden or accurate result, but a range of acceptable ones and (ii) robustness to input noisy data. For example, multimedia applications (e.g., video coding, audio filtering, image processing, and so on), highly demanded by current portable devices, are intrinsically related with human senses. The multimedia signals are in fact approximation-tolerant applications, since in [10] is stated that human senses process analog information and have difficulty to realize the negative impact of digital approximations. In other words, it is possible to adopt approximate computing techniques to improve energy efficiency in multimedia applications by adequately exploring the user experience at different profiles of quality.

II. SAD, Approximate Adders and Parallel Prefix Adders

A. SAD - Sum of the Absolute Differences

In this work we will explore: (I) the SAD sum tree with the use of approximate adders, ETA-I and Copy Adder; (II) the precise part of the approximated adders with the use of logarithmic delay adders found in the literature.

Sum of the Absolute Differences (SAD) is the metric most used to determine the greatest similarity between two video blocks in the Motion Estimation process. This metric can also be applied to determine the best offset for Fractional Motion Estimation and to choose the best Intra prediction mode. The SAD is calculated by the sum of absolute differences pixel by pixel between two blocks: the block to be coded and a reference block.

The hardware architecture of SAD consists of subtractors, absolute operators, an addition tree and an accumulator to calculate the final value of the SAD, as we will demonstrate in the equation (1), where O is the original block, R is the reference block and m and n are the dimensions of the blocks in samples. For high performance, many SAD hardware units are used in parallel within motion estimation architectures.

$$SAD = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |O_{i,j} - R_{i,j}| \tag{1}$$

Figure 1 demonstrates the SAD architecture that consists of eight subtractors, to subtract eight samples from the original block (Orig) with eight samples from the reference block (Ref), generating the residual samples that are stored in eight registers. Eight operators for absolute calculation (module) receive the input as input and perform this operation. A tree of adders accumulates the eight absolute values and generates a partial SAD value. The partial SAD is stored in a register in each clock cycle. Therefore, after eight clock cycles the final SAD value is calculated for a block of 8x8 samples.

B. Approximate Adders

Approximate computation is useful in image processing, as the human eye does not perceive some imperfections in the images. According to [8] in images there are several sources of tolerance: (I) percent limitations that are determined by the human brain’s ability to fill in missing information; (II) redundant input data, this redundancy means that an algorithm can be lossy and still be sufficient; (III) noise inputs.

The approximate adders can be classified as computational performance- and power-oriented designs. The former is related to adders divided into m independent blocks or sub-adders to speed up the critical path timing. The claim is that, for random and uniformly distributed pairs of operands, more extended carry propagation rarely occurs. Based on that, additional logic is necessary to speculate carry-in for each sub-adder, since this class of approximate adder breaks the carry propagation in many parts. Examples of adders which improve computational performance are the Error Tolerant Adder II [11], Error Tolerant Adder IV [12], and the Almost Correct Adder [13]. This class of approximate adders is also characterized by the presence of infrequent and high magnitude sum errors. Therefore, the works in [14], [15], [13], [16] proposed accuracy configurable adders to cope with this error characteristics. On the other hand, more logic is added to detect and correct the sum errors.

A different philosophy is to propose power-oriented adders which generally are divided into two parts: (i) the least significant approximate part and (ii) the most significant accurate part. Examples of power-oriented approximate adders can be observed in [17], [18], [10]. The principal idea in the approximate part is to replace the full adder cells by simpler adder circuits. Therefore, power reduction is the main objective of this class of adders. In addition, these adders also tend to reduce critical path timing, because in the approximate part there is no carry propagation scheme. One can observe that the classical truncation is a type of power-oriented adder which truncates least significant full adder cells. This class of approximate adders is also characterized by the presence of frequent and low magnitude sum errors. Such errors are of low magnitude because the bit-width of the approximate part can be controlled through an approximate parameter k. In this work, the proposed approach is to explore the power-oriented adders to give priority to power-efficiency. This is also ratified in [19] which states that the adders focused on delay reduction cannot be used to explore power-efficiency in structures which demand the massive use of additions like the multipliers. In addition, the power-oriented approximate adders enable the exploration of multiple conventional adder topologies in the precise part, which is not true for the approximate adders divided into many blocks.

Therefore, we consider in this study the exploration of the fifth approximate version of AMA (Approximate Mirror Adders) [18] and the Error Tolerant Adder I [10], because they are also explored by related works [20], [21]. The former approximate adder is renamed to “Copy adder” due to its copy function implemented by the buffers. The approximate adders which are explored in this study can be observed in Figure 3.

The “Copy adder” in Figure 3 (a) has its k bits long approximate part implemented by buffers to copy the operand a to the sum. This procedure has 50% probability of getting the correct sum for each bit position. Furthermore, the carry-in estimation for the precise part is implemented by the simpler assignment of the input operand bit b_{k-1}. This procedure has 75% probability of getting a correct carry-in estimation to the precise block. The approximate part of the ETA-I in Figure 3 (b) is implemented by the use of half adders. The sum is performed in the non-conventional direction, (*i.e.* from the most significant bit $k-1$ to the least significant bit position 0). The control logic block is conceived as follows: when the first carry-generate c is equal to "1", then all the remaining least significant sum bits are set to "1". Otherwise, the sum result is the one computed by the propagate signal. The carry-in to the precise part in ETA-I is statically set to "0". This procedure has 50% probability of getting the correct carry-in to the precise part. One can observe that for both the approximate adders, any conventional adder topology can be implemented in the precise part. Related works in [20] and [21] explore the use of the RCA. In this work, we explore the RCA plus high-performance PPA's. That is why in the next subsection a brief PPA overview is developed.

C. Precise Adders

As previously mentioned, adders are fundamental building blocks in a great variety of computational applications. Based on that, many adder topologies have been proposed to deal with the tradeoff between power and computational performance. The Ripple Carry Adder (RCA) topology is characterized to present low values of power consumption, area, and computational performance. Depending on the high-performance application requirements and given that computational complexity is increasing in nowadays tasks, the key approach is to accelerate the adder's critical path delay (*i.e.* carry propagation) at the expense of higher area and power dissipation. Based on that, the Parallel Prefix Adders (PPA's) were proposed to deal with high-performance demands [22].

The carry propagation structure in the PPA's is implemented by simple logic cells which tend to keep a regular connection. Based on that, the sum computation can be divided into pre-processing, prefix computation and post-process parts, as can be seen in Figure 4.

In the pre-processing part, the propagate p_i (*i.e.* $a_i \oplus b_i$) and generate g_i (*i.e* $a_i \wedge b_i$) signals are computed based on the input operands a_i and b_i. In the prefix computation stage, the carry computation is accelerated by the parallel composition of the black cells which implement the group propagate $P_{i:j}$ and

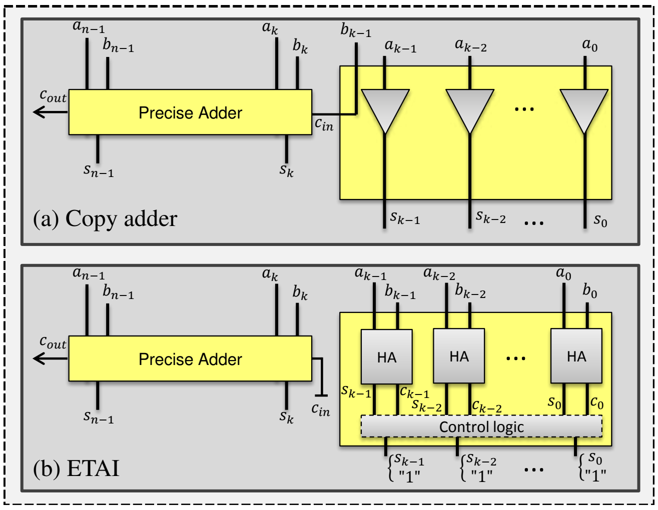

Fig. 3. Approximate adders: (a) Copy adder; (b) ETA-I.

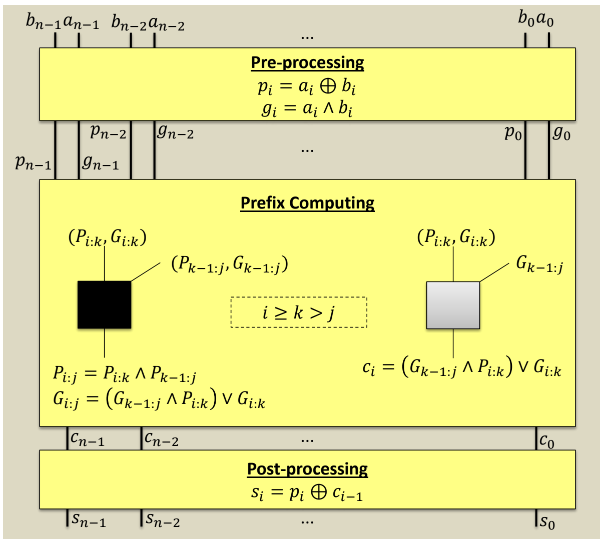

Fig. 4. Parallel Prefix Adder Structure

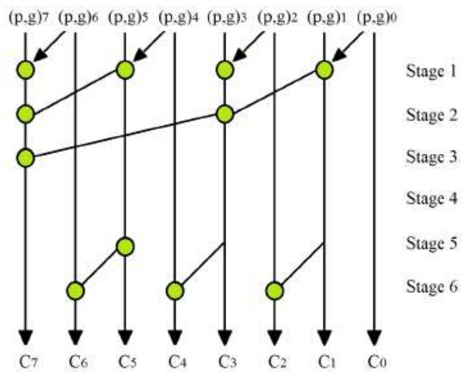

Fig. 5. Brent-Kung Adder.

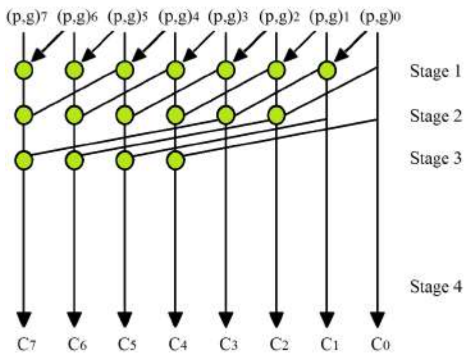

Fig. 6. Kogge-Stone Adder.

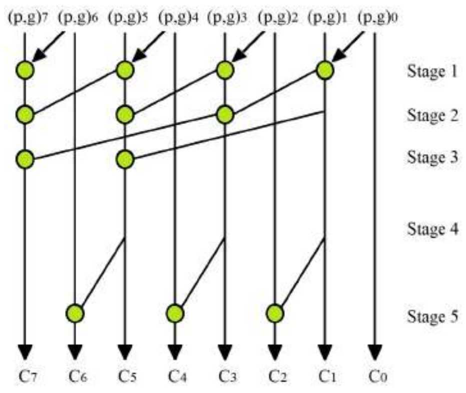

Fig. 7. Han-Carlson Adder.

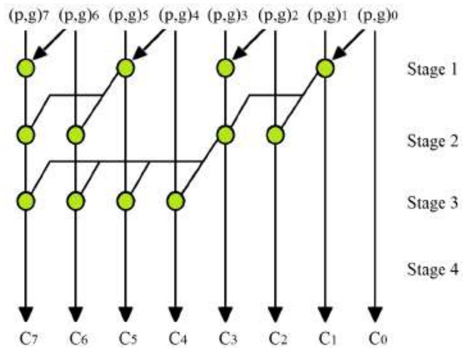

Fig. 8. Sklansky Adder.

generate $G_{i:j}$ signals as well as the grey cells which compute the carry c_i. Finally, the post-processing step is given by the sum $s_i = p_i \oplus c_{i-1}$.

1) Use of PPA adders on the precise part of the approximate adders: Different configurations between the black and grey cells can be obtained. According to the PPA's taxonomy presented in [23], the different prefix cells configurations allow tradeoffs among (i) the number of logic levels, (ii) the maximum fanout, and (iii) the maximum number of horizontal wire tracks (*i.e* wire density). All of these aspects affect the adder delay. Based on that, PPA's topologies proposed in [24], [2], [3], [5], [4] are considered in this study. Their main characteristics concerning logic levels, maximum fanout, and the maximum number of wiring tracks are presented in Table I [23].

TABLE I. TAXONOMY OF n-BIT PPA'S

PPA	logic levels	maximum fanout	maximum wiring tracks
Brent-Kung (B-K) [24]	$2\log_2(n) - 1$	2	1
Sklansky (SK) [4]	$\log_2(n)$	$\frac{n}{2} + 1$	1
Kogge-Stone (K-S) [25]	$\log_2(n)$	2	$\frac{n}{2}$
Han-Carlson (H-S) [3]	$\log_2(n) + 1$	2	$\frac{n}{4}$
Ladner-Fischer (L-F) [5]	$\log_2(n) + 1$	$\frac{n}{4} + 1$	1

As can be seen in Table I, the Brent-Kung [Figure 5] adder has the highest number of levels, while presents low values for both fanout and wire density. The Sklansky [Figure 8] and Kogge-Stone [Figure 6] have the lowest number of logic levels, but the former presents the highest fanout, and the latter has the worst wire density due to the highest number of prefix cells. The Han-Carlson [Figure 7] is the hybrid solution between the Brent-Kung and Kogge-Stone. Therefore, this adder balance the tradeoff between the number of logic levels and wiring tracks. The Ladner-Fischer [Figure 9] adder is the hybrid approach between Sklansky and Brent-Kung so

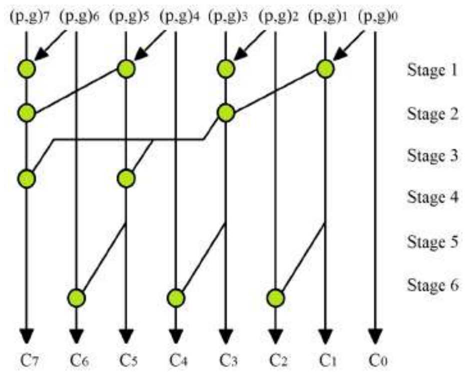

Fig. 9. Ladner Fischer Adder.

that the tradeoff is balanced between the number of logic levels and the fanout.

III. Results

The SAD metric has its sum tree explored, em 16, 24 e 32 bits, by the approximate Adders Copy and ETA-I that employ the parallel prefix adders in the precise part. All circuits were implemented in VHDL language and synthesized in 45 nm technology using the Cadence RTL Compiler tool. The approximate adders make up half the bit width of the adders for all designs under evaluation. During the synthesis, all the combinational logic of the designs is preserved to ensure that there are no optimizations or changes in the traditional adder topologies.

The power was estimated by extracting the switching activity with random numbers as input. The results referring to the area under analysis iso-performance at 500 MHz.

For the power results it was analyzed the trade-off based on the RCA adder for comparison between the PPAs cells. The choice of RCA is because it is the most commom adder and it is a not a PPA adder.

The power results were evaluated for the approximate adders Copy and ETA-I for the sum tree in the SAD architectures. Both, Copy and ETA-I adders were evaluated with the PPA adders in the addition computer precise part.

For 16, 24 and 32 bits at a frequency of 500MHz the Ladner Fische adder presented the smaller power dissipation. This adder showed an improvement over the RCA of 2.45% (Figure 10) for the 16-bit Copy adder, 1.52% (Figure 11) for the 16-bit ETA-I adder, 11.64% (Figure 12) for the 24-bit Copy adder, 9.49% (Figure 13) for the 24-bit ETA-I adder, 17.42% (Figure 14) for the 32-bit Copy adder and 16.15% (Figure 15) for the 32-bit ETA-I adder.

With the power consuption analyzes it is possible to infer that at higher clock frequency rates the PPA adders significantly reduce the power dissipation, since these adders were designed to handle large computational efforts. The Ladner Fischer adder performed the best performance by having a regularity of layout that lies between the terms of the Sklansky and Brent Kung adders, this adder calculates the prefix for odd numbers and uses one more stage to undo the even positions in the prefix computation.

[h!]

IV. Conclusion

This work proposed the exploration of the SAD sum tree, which is explored among different summing topologies to implement the precise part of approximate state-of-the-art additives. Depending on the demands of applications, one can consider using a different topology than adopting only the low-performing RCA. This work allows a more significant design space, where performance and power consumption can be optimized to generate energy efficient projects.

V. Acknowledgments

The authors thank CNPq and Capes for financial support.

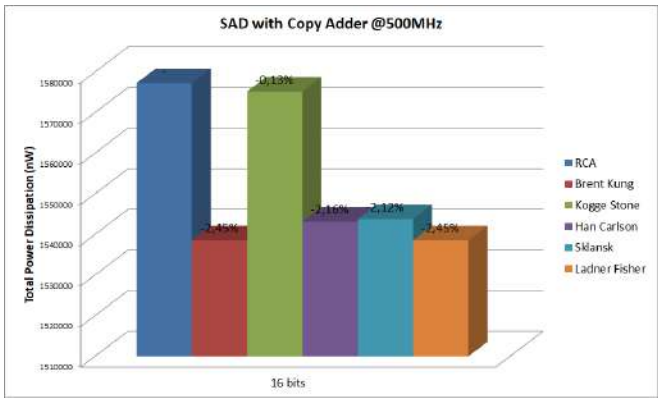

Fig. 10. Power Results SAD with Copy Adder 16 bits

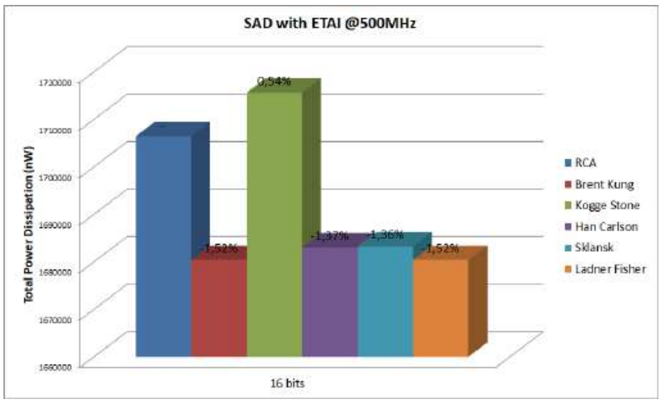

Fig. 11. Power Results SAD with ETAI 16 bits

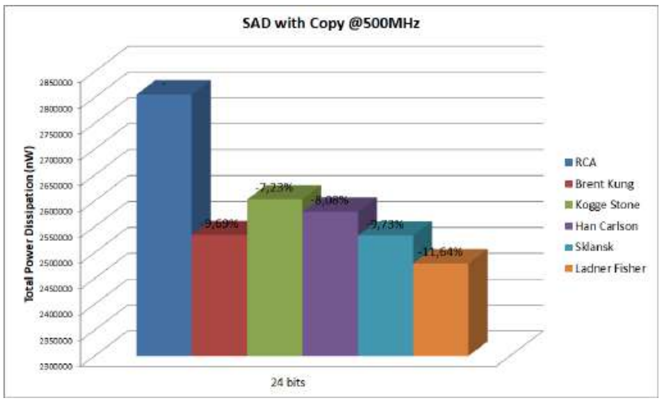

Fig. 12. Power Results SAD with Copy Adder 24 bits

References

[1] R. P. Brent and H. Kung, “A regular layout for parallel adders,” *IEEE Trans. Computer*, 1982.

[2] P. M. Kogge and H. S. Stone, “A parallel algorithm for the efficient solution of a general class of recurrence equations,” *IEEE Transactions on Computers*, vol. C-22, no. 8, pp. 786–793, Aug 1973.

[3] T. Han and D. Carlson, “Fast area-efficient vlsi adders,” in *Computer*

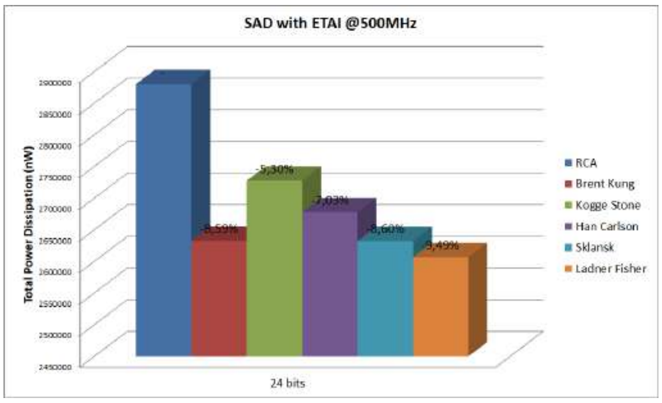

Fig. 13. Power Results SAD with ETAI 24 bits

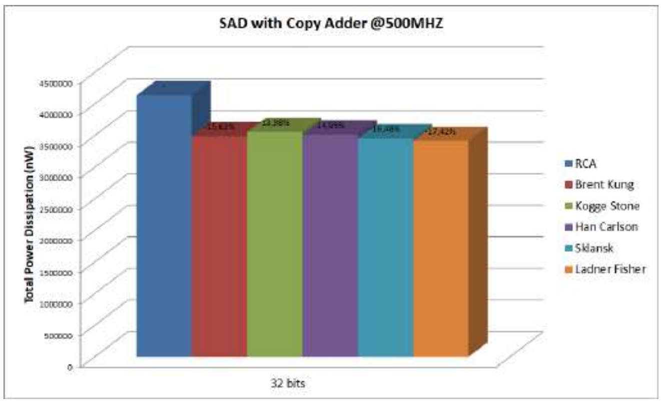

Fig. 14. Power Results SAD with Copy Adder 32 bits

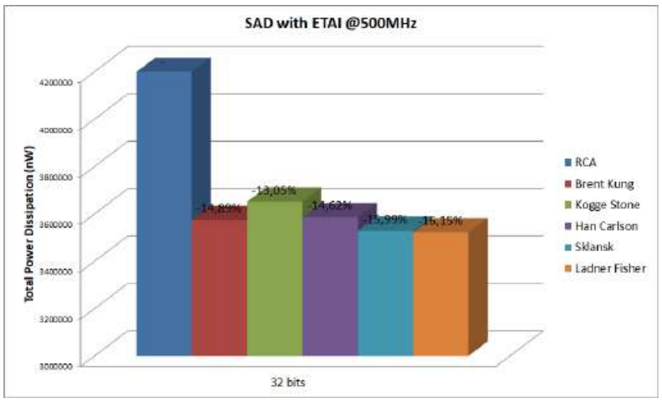

Fig. 15. Power Results SAD with ETAI 32 bits

Arithmetic (ARITH), 1987 IEEE 8th Symposium on. IEEE, 1987, pp. 49–56.

[4] J. Sklansky, “Conditional-sum addition logic,” in *IRE Transactions on Electronic Computers*, vol. 2, 1960, pp. 226–231.

[5] R. Ladner and M. Fischer, “Parallel prefix computation,” *Journal of the ACM (JACM)*, vol. 27, no. 4, pp. 831–838, 1980.

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

[7] S. Kim and Y. Kim, “Adaptive approximate adder to reduce error distance for image processor,” in *2016 International SoC Design Conference (ISOCC)*, 2016.

[8] J. Han and M. Orshansky, “Approximate computing: An emerging paradigm for energyefficient design.” in *IEEE European Test Symposium*, no. 18, Avignon, 2013, pp. 1–6.

[9] S. Venkataramani, S. T. Chakradhar, K. Roy, and A. Raghunathan, “Approximate computing and the quest for computing efficiency,” in *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, June 2015, pp. 1–6.

[10] N. Zhu, W. Goh, W. Zhang, K. Yeo, and Z. Kong, “Design of low-power high-speed truncation-error-tolerant adder and its application in digital signal processing,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 8, pp. 1225–1229, 2010.

[11] N. Zhu, W. Goh, and K. Yeo, “An enhanced low-power high-speed adder for error-tolerant application,” in *12th International Symposium on Integrated Circuits, ISIC09*, 2009, pp. 69–72.

[12] N. Zhu, W. Goh, G. Wang, and K. Yeo, “Enhanced low-power high-speed adder for error-tolerant application,” in *International SoC Design Conference (ISOCC)*. IEEE, 2010.

[13] A. K. Verma, P. Brisk, and P. Ienne, “Variable latency speculative addition: A new paradigm for arithmetic circuit design,” in *2008 Design, Automation and Test in Europe*, March 2008, pp. 1250–1255.

[14] M. Shafique, W. Ahmad, R. Hafiz, and J. Henkel, “A low latency generic accuracy configurable adder,” in *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, June 2015, pp. 1–6.

[15] A. B. Kahng and S. Kang, “Accuracy-configurable adder for approximate arithmetic designs,” in *DAC Design Automation Conference 2012*, June 2012, pp. 820–825.

[16] R. Ye, T. Wang, F. Yuan, R. Kumar, and Q. Xu, “On reconfiguration-oriented approximate adder design and its application,” in *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov 2013, pp. 48–54.

[17] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, “Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 4, pp. 850–862, April 2010.

[18] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, “Low-power digital signal processing using approximate adders,” in *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 1, 2013, pp. 124–137.

[19] S. Rehman, W. El-Harouni, M. Shafique, A. Kumar, J. Henkel, and J. Henkel, “Architectural-space exploration of approximate multipliers,” in *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov 2016, pp. 1–8.

[20] J. Oliveira, L. Soares, E. Costa, and S. Bampi, “Exploiting approximate adder circuits for power-efficient gaussian and gradient filters for canny edge detector algorithm,” in *IEEE LASCAS*, 2016, pp. 379–382.

[21] Y. Kang, J. Kim, and S. Kang, “Novel approximate synthesis flow for energy-efficient FIR filter,” in *2016 IEEE 34th International Conference on Computer Design (ICCD)*, Oct 2016, pp. 96–102.

[22] A. Beaumont-Smith and C.-C. Lim, “Parallel prefix adder design,” in *IEEE Symposium on Computer Arithmetic*, 2001, pp. 218–225.

[23] D. Harris, “A taxonomy of parallel prefix networks,” in *The Thirty-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, vol. 2, Nov 2003, pp. 2213–2217 Vol.2.

[24] R. Brent and H.-T. Kung, “A regular layout for parallel adders,” *IEEE Transactions on Computers*, vol. 31, no. 3, pp. 260–264, 1982.

[25] P. Kogge and H. Stone, “A parallel algorithm for the efficient solution of a general class of recurrence equations,” *IEEE Transactions on Computers*, vol. 100, no. 8, pp. 786–793, 1973.

Morgana Macedo Azevedo da Rosa is currently a Computer Engineering student at the Catholic University of Pelotas (UCPel), Pelotas, Brazil. Her research interests are Arithmetic Circuits and VLSI design.

Bianca Santos da Cunha da Silveira she holds a degree in Electronic Engineering from the Catholic University of Pelotas (2008) and a Master's degree in Electronic Engineering and Computation from Catholic University of Pelotas (2016), working mainly on the following topics: video coding, microelectronics, hardware architectures design, VHDL.

Leonardo Bandeira Soares (S'12) received the five-year engineering degree in Computer Engineering from the Federal University of Rio Grande, Rio Grande, Brazil, in 2010, and the M.Sc. and Ph.D. degree in Microelectronics from the Federal University of Rio Grande do Sul, Porto Alegre, Brazil, in 2013 and 2018, respectively. He is currently professor at the Federal Institute of Technology of Rio Grande do Sul (IFRS). His research interests are VLSI architectures, Approximate Computing, Video Coding, Digital Signal Processing, and Energy-Efficiency in CMOS design.

Cláudio Machado Diniz (S'08–M'15) received the Computer Engineering degree from Federal University of Rio Grande (FURG), Brazil (2007) and the M.Sc. and Ph.D. degrees in Computer Science from Federal University Rio Grande do Sul (UFRGS), Brazil (2009 and 2015). He is currently an Assistant Professor at the Catholic University of Pelotas (UCPel), Pelotas, Brazil. He has worked as intern researcher at the Chair for Embedded Systems (CES) of Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany. His research interests include image and video processing algorithms, architectures and VLSI design.

Eduardo Antonio Ceśar da Costa (M'13) received the five-year engineering degree in Electrical Engineering from the University of Pernambuco, Recife, Brazil, in 1988, the M.Sc. degree in electrical engineering from the Federal University of Paraiba, Campina Grande, Paraiba, Brazil, in 1991, and the Ph.D. degree in computer science from the Federal University of Rio Grande do Sul, Porto Alegre, Brazil, in 2002. Part of his doctoral work was developed at the Instituto de Engenharia de Sistemas Computadores (INESC-ID), Lisbon, Portugal. He is currently a Full Professor at the Catholic University of Pelotas (UCPel), Pelotas, Brazil. He is co-founder and coordinator of the Graduate Program on Electronic Engineering and Computing at UCPel. His research interests are VLSI architectures and low-power design.