Power-Efficient Approximate SAD Architecture with LOA Imprecise Adders

Roger Porto, Luciano Agostini, *Senior Member, IEEE*, Bruno Zatt, *Senior Member, IEEE*,
Nuno Roma, *Senior Member, IEEE and* Marcelo Porto, *Senior Member, IEEE*

Abstract — **Approximate computing is a highly promising approach to reduce the computational effort in video encoders. Its use is even more relevant and advantageous when high resolution videos must be processed in real time using battery powered devices. In this scenario, it is essential to reduce power dissipation and silicon area. In particular, the distortion metric calculation module is one of the most time demanding and the Sum of Absolute Differences (SAD) is usually the most used distortion metric, mainly when dedicated hardware is considered. To overcome this demand, this paper presents a power-efficient SAD architecture compliant with current video encoders based on the usage of Lower-Part-OR Adders (LOA). The attained results showed that important power (17.99%) and area (30.56%) savings can be reached, with an increase of only 0.3% in BD-rate. When compared with state of the art related works, the designed architecture reaches the best area and power dissipation results.**

Index Terms— **Approximate computing, imprecise adders, SAD, video coding, power efficiency.**

I. Introduction

VIDEO coding is a key technology for current multimedia applications, since high spatial and temporal resolution videos use contents with support to 3D representation are becoming more and more common in this application domain. On the other hand, there is nowadays an important academic and industrial effort to increase the video coding efficiency (bitrate versus image quality). The state-of-the-art High Efficiency Video Coding (HEVC) standard [1], the Chinese Audio Video Standard 2 (AVS2) [2], the recently released AV1 encoder from Alliance for Open Media (AOM) [3] and the Versatile Video Coding (VVC) [4], which will be the next generation of ISO and IEC standard, are examples of this effort.

On the other hand, multimedia applications are increasingly migrating to battery-powered devices, such as smartphones, digital cameras, virtual reality appliances and others. In this scenario, not only the video coding efficiency must be highly considered, but the encoding throughput, the power dissipation, the energy consumption, and the system cost must be regarded as fundamental key aspects. In fact, considering the very high computational effort that is already required for current video encoders, dedicated hardware designs have been extensively used to allow a power-efficient real-time processing of videos.

To overcome this demand, approximate computing is regarded as a highly promising approach to design energy-efficient hardware video codecs [5]. This approach exploits the characteristics of error-tolerant applications, i.e., applications that are resilient to numerically imprecise partial results. In fact, by tolerating a minor loss of accuracy, it is possible to achieve substantially improved energy efficiency [5]. Moreover, it has been shown that the introduction of a limited amount of approximate computing in video coding algorithms often results in almost imperceptible visual artifacts [6], mostly because of the error-tolerant characteristics of the adopted encoding tools.

The Sum of Absolute Differences (SAD) is one of the most commonly used distortion metrics in current video encoders, especially those designed in hardware. For many current encoders, SAD is used in all video coding prediction steps (such as interframe, intraframe and interview predictions). This makes the SAD calculation a crucial operation to define the encoder hardware efficiency. On the other hand, the prediction steps that use this operator are naturally resilient to small imprecisions, since there are many prediction options that may conduct to similar encoding efficiency results. This makes the SAD calculation a rather promising operation to apply approximate computing, in order to decrease the encoder area and power dissipation, with minimal impacts at the encoding efficiency.

Some published works present SAD architectures targeting different technologies and with focus on a diversity of applications and video encoders. The works [7], [8], [9], and [10] target standard cells implementations. Only [8] explores imprecision to reduce the SAD computation cost, using approximations over the most significant bits.

This paper proposes a novel power-efficient SAD calculation architecture using imprecise Lower-Part-OR Adders (LOA) [11].

R. Porto, L. Agostini, B. Zatt, and M. Porto are with the Video Technology Research Group (ViTech), Federal University of Pelotas, 96010-610 Pelotas, Brazil (email: recporto@inf.ufpel.edu.br; agostini@inf.ufpel.edu.br; zatt@inf.ufpel.edu.br; porto@inf.ufpel.edu.br).

N. Roma is with INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, 1000-029 Lisbon, Portugal (email: nuno.roma@inesc-id.pt).

978-1-7281-0453-9/19/$31.00 ©2019 IEEE

II. Imprecise Arithmetic Operators

Arithmetic operators can significantly influence the overall performance of a digital system [12]. In fact, arithmetic operators are not only the main responsible for the delay but are also the main cause of power dissipation in digital circuits, mostly due to carry propagation chains [13]. Approximate computing approaches are usually used to mitigate this problem, by shortening or truncating the carry chain. The resulting operator becomes faster, and its power dissipation gets reduced, at the cost of the introduction of a certain level of imprecision in the results.

Several imprecise adders have been proposed in the literature. Some examples are the Accuracy-Configurable Adder [14], Carry Cut-Back Adder [15], Error-Tolerant Adder [16] [17], Generic Accuracy Configurable Adder [18], Lower-Part-OR Adder [11], among others.

The Accuracy-Configurable Adder (ACA-II) [14] and the Carry Cut-Back Adder (CCB) [15] are segment-based imprecise adders. While the first employs three overlapping sub-adders to reduce the carry chain, the CCB cuts the carry propagation chain at lower-significance positions, using the carry propagation only at the high-significance stages.

Error-Tolerant Adders (ETA) are proposed in [16] and [17]. The first one is the ETA-I [16], classified as an approximate full adder. This operator splits the addition in two, by applying only imprecise techniques in the lower part, where all bits are checked from left to right. The ETA-IV [17] is also a segment-based approximate adder with non-overlapping sub-adders. Specialized units generate the carries from one stage to another.

In [18], a Generic Accuracy Configurable Adder (GeAr) is proposed. A GeAr adder can be configured by specifying the number of sub-adder units and, for each sub-adder, the number of prediction bits, the number of sum bits and the bit width. Once again, overlapped areas are used.

The Lower-Part-OR Adder (LOA) [11] splits the addition into two smaller parts. The upper-part performs regular precise addition with the most significant bits. On the other hand, the computation of the least significant bits is simplified, and the carry chain propagation is eliminated (as depicted in Fig. 1) by applying bitwise OR to the inputs. An extra AND gate is used in the most significant bits of the imprecise part to generate a carry-in for the upper-part, in order to decrease the imprecision [11].

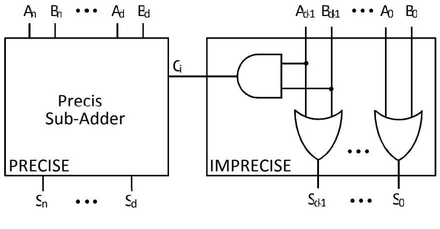

Fig. 1. Lower-Part-OR Adder structure.

TABLE I
BD-Rate Results for the Considered Adder Structures

Adder	ACA-II	CCB	ETA-I	ETA-IV	GeAr	LOA
BD-rate (%)	4.9	1.6	1.3	0.3	2.6	0.3

III. Imprecise Addition Evaluation in SAD

To identify the most suitable imprecise adders configurations to be adopted by the video encoder SAD computation module, a set of 8-bit imprecise addition functions were described in C++ and evaluated using 99,842 input values extracted from actual video sequences. A total of 26 versions of the six 8-bit previously presented operators were evaluated. Through these first analyses, the best configurations for these imprecise adders were defined as: (i) ACA-II with three overlapping sub-adders, four bits each; (ii) CCB with two sub-adders with four bits each and one bit at the cut-back definition; (iii) ETA-I with three precise and five imprecise bits; (iv) ETA-IV with three sub-adders (three, three and two bits) using two and three bits in the first and second carry generations, respectively; (v) GeAr using two overlapping sub-adders with five bits each; and (vi) LOA with three precise and five imprecise bits.

Then, the six identified imprecise adders were embedded in the motion estimation encoding tool of the HM 16.12 HEVC reference software, to evaluate their impact on the coding efficiency (in terms of BD-rate). The imprecise operators were inserted only in the first stage of the SAD operations in order to avoid accumulated error effects. This first operation corresponds to the subtraction that is needed to generate the SAD absolute differences. The results of this evaluation are presented in Table I and they were obtained by encoding the twenty test video sequences and the four QP values recommended in the Common Test Conditions (CTC) [19] and using the Low Delay P Main configuration. According to Table I, the best results were obtained for LOA and ETA-IV.

The next experiment was conducted to obtain a characterization of the imprecise adders in hardware. The operators were described in VHDL and synthesized for a 45nm@1.1V Nangate standard cell library and using the Cadence Encounter RTL Compiler tool. These experimental results were compared with those obtained for the ripple-carry (RCA) precise adders, by considering the increase (or decrease) in terms of: (i) power dissipation, (ii) delay, (iii) area, and (iv) power delay product (PDP). This multi-variable comparison of the imprecise operators is depicted in Fig. 2 in the form of radar charts. The BD-rate results were also included. In Fig. 2, the smaller the gray area is, the better is the solution. One can notice that LOA operator reached the best results, since it presented the smallest gray area.

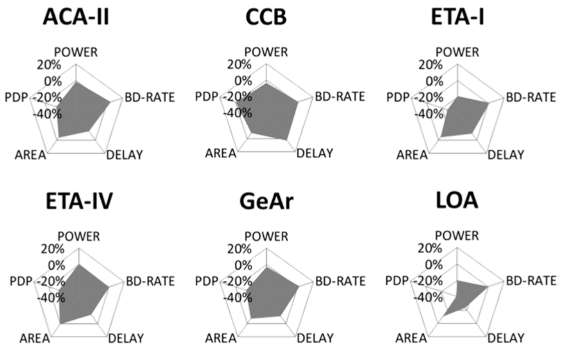

Fig. 2. Multi-variable comparison of several imprecise adders.

IV. Approximate Computing SAD Architecture

The SAD architecture proposed in this paper aims the processing of all the supported block sizes of current video encoders. Three distinct design options can be considered to attain this objective. The first one is the usage of an iterative solution, supported on an accumulation mechanism to conduct as many iterations as necessary to generate the results, as proposed in [9]. The second option is to use as many instances of this architecture as necessary to process all available block sizes, by accumulating the parallel partial results and by reusing SAD calculations to generate the SAD for different block sizes, as proposed in [20]. Finally, a mixture of both alternatives could also be used, by scaling the parallelism to reach the throughput required by the application.

The work in [20] presents a previous investigation about the use of imprecision in SAD calculations, by using a SAD unity able to process four comparisons per cycle and organized in a pipeline with three stages. Such work also investigates the use of LOA operator with different imprecision levels applied in the variable block size motion estimation (VBSME) process.

In contrast, the SAD architecture that is now proposed significantly extends such preliminary study by implementing 16 simultaneous comparisons per cycle in a fully combinational scheme, being able to process all the supported block sizes of current video standards

Fig. 3 presents a diagram with the arithmetic operations performed by the SAD calculation architecture. The architecture is divided in five addition levels. The first level delivers the absolute difference values between the samples from the original block (ORGn inputs) and the current block (CURn inputs). An adder tree formed by the next levels adds the partial values coming from the first level. The imprecise LOA operators were used in the first level (as discussed before), while the other levels use RCA. To ensure a reliable comparison, an alternative version of this architecture, without imprecision, was also designed.

V. Synthesis Results

Both the precise and imprecise SAD architectures were synthesized with the Cadence Encounter RTL Compiler tool using a 45nm@1.1V Nangate standard cell library. The target frequency was defined to 300MHz, intending to process video sequences with 1920x1080 at 30 frames per second using, as reference, the evaluation presented in [7] without any encoder restriction. The power evaluation considered 100,000 real stimuli gathered from video samples extracted from the HM 16.12 reference software. Table II summarizes the synthesis results of these two architectures, by considering both power and area.

TABLE II
Synthesis Results

	Precise	Imprecise	Savings
Power (mW)	0.417	0.342	17.99%
Area (gates)	1,623	1,127	30.56%

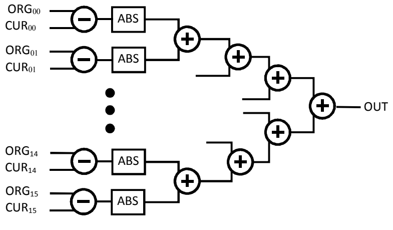

Fig. 3. SAD calculation architecture.

The obtained results show that the use of LOA causes significant impacts in terms of power dissipation and silicon area. The reached power savings of the imprecise architecture were as high as 17.99%, when compared with the precise version. The circuit area savings were even more impressive: 30.56%. It is important to emphasize that these power and area savings were reached with a negligible degradation of only 0.3% in BD-rate. Considering the target frequency of these syntheses (300MHz) the architecture can perform 4.8 billion of comparisons/second. This throughput can still be increased by using other instances of this architecture (increasing the parallelism), or even by increasing the target synthesis frequency.

VI. Comparison with Related Works

Table III presents a comparison of the proposed architecture with related works. Since these works considered rather different technologies, operating frequencies and parallelism levels, it is difficult to conduct a complete and fair comparison. To realize a fairer comparison, an additional synthesis was performed by targeting the 500MHz operating frequency (the same as the related work in [9]). Even when operating at such higher frequency, our architecture uses the lowest silicon area and presents a lower power dissipation when compared with all related works.

The works [7] [9] [10] did not use any SAD solution that cause coding efficiency degradation. The work [8] uses approximations in the SAD calculations to reduce the complexity, but the BD-rate results are not available. The work in [7] is the only one that used the same technology of our implementation, but with half of the parallelism supported by our architecture. Even supporting a higher throughput at the same operation frequency, our architecture reaches almost three times lower power dissipation and requires 38% less hardware than [7]. When compared with [8], our solution reached a power dissipation almost 12 times lower, when running at a 62% higher frequency. However, it is worth to note that the technology used in [8] is older, highly influencing this result. The work in [9] presents an

TABLE III
Comparisons with Related Works

Work	Technology	Freq. (MHz)	Samples/ cycle	Area	Power (mW)
This	**Nangate 45nm**	**300**	**16**	**1,127 gates**	**0.34**
This	**Nangate 45nm**	**500**	**16**	**1,127 gates**	**0.43**
[7]	Nangate45 nm	300	8	1,838 gates	0.97
[8]	Samsung 130nm	185	16	0.015mm^2	4.18
[9]	TSMC 180nm	504	16	5,778 gates	3.79
[10]	GF 180nm	345	16	0.02 mm^2	-

architecture also targeting an older technology. Even so, our architecture used about five times less gates and has a power dissipation that is almost nine times lower than [9]. The work in [10] did not present power results and the area results are presented in mm^2, making it impossible to reliably compare the results. It is important to emphasize that, in some cases, the compared works include additional modules in the presented designs, such as accumulators and registers, which impact in the power and area results.

Finally, an important aspect that should be noted is the fact that precise SAD architectures typically reach lower throughputs and higher power consumptions. As a consequence, encoder hardware designs using precise SAD architectures after include some encoder restriction to allow the processing of high resolution videos in real time. As a result, even when using a precise SAD architecture, the required encoder restrictions often lead to a global coding efficiency degradation. For example, [7] defines the target operation frequency avoiding some encoding modes to allow the desired power/throughput operation point, leading to a 0.4% average coding efficiency loss (BD-rate). This means that even using a precise SAD version, [7] results in higher coding efficiency degradation when compared to our proposal (0.3%). Other works also consider this type of approach, where the encoder restrictions lead to significant coding efficiency degradations.

VII. Conclusions

This paper presented a power-efficient approximate SAD architecture using imprecise LOA adders. The architecture performs 16 comparisons per cycle in parallel and can be unfolded to reach different throughput requirements. To support the proposed design, an evaluation of different imprecise adders was performed; it was concluded that LOA is the best option in the designed SAD architecture. Furthermore, these experiments show that the imprecise operator should be used only in the first stage of SAD calculations, in order to minimize the encoding efficiency losses. The implemented SAD architecture offers a power reduction of 17.99% when compared with an architecture based only on RCAs. The area savings are even more expressive, reaching 30.56% of reduction by comparing to the RCA version. On the other hand, the use of LOA in the first SAD calculation stage implied a negligible impact in BD-rate (0.3%). When compared with other related works, the designed approximate SAD architecture reaches the best area and power dissipation results. Finally, it is important to emphasize that the designed SAD architecture is compliant with any currently existing video encoders that use SAD as distortion criteria.

Acknowledgment

This work is partly financed by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) Finance Code 001, by FCT projects PTDC/EEI-HAC/30485/2017 and UID/CEC/50021/2013, and also by CNPq and FAPERGS Brazilian research support agencies.

References

[1] G. J. Sullivan et al., “Overview of the high efficiency video coding (HEVC) standard”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.

[2] Z. He et al., “AVS2-video coding standard - An application-oriented and high performance video coding standard”, in *Proc. of ICME 2014*, Chengdu, China, 2014, pp. 1-5.

[3] Alliance for Open Media, <http://aomedia.org/>

[4] Developing a video compression algorithm with capabilities beyond HEVC [Online]. https://www.itu.int/en/ITU-T/studygroups/2017-2020/16/Pages/video/jvet.aspx, Accessed on: Aug., 2018.

[5] J. Han, and M. Orshansky, “Approximate computing: an emerging paradigm for energy-efficient design”, in *Proc. of ETS 2013*, Avignon, France, 2013, pp. 1-6.

[6] A. Raha et al., “A power efficient video encoder using reconfigurable approximate arithmetic units”, in *Proc. of VLSID 2014*, Mumbai, India, 2014, pp. 324-329.

[7] B. Silveira et al., “Power-efficient sum of absolute differences hardware architecture using adder compressors for integer motion estimation design”, *IEEE Transactions on Circuits and Systems-I: Regular Papers*, pp. 1-12, Aug. 2017.

[8] L. Dinh et al., “a-SAD: power efficient SAD calculator for real time H.264 video encoder using MSB-approximation technique”, in *Proc. of IEEE/ACM ISLPED 2014*, La Jolla, USA, 2014, pp. 259-262.

[9] C. Diniz et al., “Comparative analysis of parallel SAD calculation hardware architectures for H.264/AVC video coding”, in *Proc. of LASCAS 2010*, Iguaçu Falls, Brazil, 2010, pp. 113-116.

[10] N. Vayalil et al., “ASIC design in residue number system for calculating minimum sum of absolute differences”, in *Proc. of ICCES 2015*, Cairo, Egypt, 2015, pp. 129-132.

[11] H. Mahdiani et al., “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, 2010.

[12] F. Frustaci et al., “Designing high-speed adders in power-constrained environments”, *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 56, no. 2, pp. 172-176, 2009.

[13] S. Dutt, S. Nandi, and G. Trivedi, “A comparative survey of approximate adders”, in *Proc. of RADIOELEKTRONIKA 2016*, Kosice, Slovakia, 2016, pp. 61-65.

[14] A. Kahng, and S. Kang, “Accuracy-configurable adder for approximate arithmetic designs”, in *Proc. of DAC 2012*, San Francisco, USA, 2012, pp. 820-825.

[15] V. Camus, J. Schlachter, and C. Enz, “A low-power carry cut-back approximate adder with fixed-point implementation and floating-point precision”, in *Proc. of DAC 2016*, Austin, USA, 2016, pp. 1-6.

[16] N. Zhu et al., “Design of low-power high-speed truncation-error-tolerant adder and its application in digital signal processing”, *IEEE Transactions on Very Large Scale Integration Systems*, vol. 18, no. 8, pp. 1225- 1229, 2010.

[17] N. Zhu, W. Goh, G. Wang, and K. Yeo. “Enhanced low-power high-speed adder for error-tolerant application”, in *Proc. of ISOCC 2010*, Incheon, Korea, pp. 323-327.

[18] M. Shafique, W. Ahmad, R. Hafiz, and J. Henkel, “A low latency generic accuracy configurable adder”, in *Proc. of DAC 2015*, San Francisco, USA, 2015, pp. 1-6.

[19] F. Bossen, “Common test conditions and software reference configurations”, document JCTVC-L1100 of JCT-VC, 2013.

[20] R. Porto et al., “Energy-efficient motion estimation with approximate arithmetic”, in Proc. of MMSP 2017, Luton, England, 2017, pp. 1-6.