Energy-Efficient Motion Estimation with Approximate Arithmetic

Roger Porto, Luciano Agostini, Bruno Zatt, Marcelo Porto
Video Technology Research Group (ViTech)
Center of Technological Development (CDTec)
Federal University of Pelotas (UFPel)
Pelotas, Brazil
{recporto, agostini, zatt, porto}@inf.ufpel.edu.br

Nuno Roma, Leonel Sousa
INESC-ID, Instituto Superior Técnico (IST)
Universidade de Lisboa
Lisboa, Portugal
{nuno.roma, leonel.sousa}@inesc-id.pt

***Abstract*—Energy efficiency has become a primary concern in the design of multimedia digital systems, particularly when targeting mobile devices. Approximate computing is a highly promising approach to address this challenge. This paper presents an architectural exploration in a variable block size motion estimation (VBSME) architecture using imprecise Lower-Part-OR Adders (LOA). These adders were applied to Sum of Absolute Differences units (SAD) in order to reduce the energy consumption while introducing a minimum impact on the coding efficiency. Three VBSME architectures with LOA operators were developed by considering different imprecision levels. The conducted evaluations, performed using the High-Efficiency Video Coding standard (HEVC) reference software, showed that this technique introduces a negligible impact on the coding efficiency (between 0.6% and 2.5% increase of the BD-Rate). Nevertheless, when the designed architectures were synthesized for a 45nm standard cells technology, significant power savings were observed (between 7% and 11.5%, depending on the used LOA version), demonstrating the viability and significant gains of the proposed approach.**

Keywords—approximate computing; approximate adders; motion estimation; low power design; video coding.

I. Introduction

The High Efficiency Video Coding standard (HEVC) approximately doubles the coding efficiency when compared with its predecessor, the H.264/AVC standard [1]. To provide such increased performance, video encoders implement a considerable number of high complexity digital signal processing algorithms. Thus, dedicated hardware architectures are becoming mandatory to provide a good trade-off between power consumption and coding efficiency.

Multimedia applications and devices are becoming more and more mobile. Thus, energy efficiency becomes a significant concern. In this way, a promising approach to energy-efficient design of digital systems is the usage of approximate computing [2][3]. Several low-power design techniques are related to this paradigm. This approach is based on the concept of error-tolerant applications [4], i.e., the applications which are resilient to numerically imprecise partial results. Thus, by tolerating a minor loss of accuracy, it is possible to achieve substantially improved energy efficiency [2]. So, the error-resilience of applications is the main motivation behind the use of approximate computing.

Besides being one of the most important multimedia tools, video coding is an example of application that can be improved, in terms of energy efficiency, by inserting approximate computing techniques. The introduction of a limited amount of approximate computing in the video coding algorithms often results in almost imperceptible visual artifacts [5] due to the limitations of the human visual system (HVS) [6]. Thus, employing approximate computing on dedicated hardware video encoders is a promising strategy for energy reduction.

Motion Estimation (ME) stands out in this context because it is one of the most complex and energy demanding operation inside a video encoder. Multiple hardware solutions for ME are found in the literature, such as [7], [8], [9], [10], and [11]. Although diverse settings are used, none of them take advantage of approximate operators.

Nevertheless, ME presents a high degree of resilience for small arithmetical errors. Since ME is basically a search for the block of the previously processed frames that is the most similar to the block in analysis, the choice of a non-optimal computation does not cause any inconsistence in the encoder process. In fact, most video encoders actually use fast ME algorithms, which significantly reduces the ME complexity, at a cost of a non-optimal result, causing a minor degradation of the encoding efficiency. By following this approach, and despite the encoder tools dependences, most published works propose several techniques to reduce the global complexity through the reduction of the encoder tools local complexity. This strategy is particularly used for the hardware designs and algorithmic optimizations targeting the ME.

Accordingly, this paper presents an energy-efficient variable block size motion estimation hardware design, called E-VBSME, which uses imprecise Lower-Part-OR Adders (LOA) to reduce energy consumption with minimal impacts in the encoding efficiency. The LOA operators were inserted in the SAD calculations, by replacing some of the original operators. To evaluate the impacts of this approach, the E-VBSME architectures were designed by considering a 45nm standard cells technology and by targeting real-time processing of high definition videos using the HEVC standard [12].

The paper is organized as follow. The next section presents the LOA definition. Section III proposes the energy efficient VBSME architectures. Section IV presents a software

This work was supported by CAPES, CNPq and FAPERGS Brazilian agencies and by FCT Portuguese agency.

978-1-5090-3649-3/17/$31.00 ©2017 IEEE

evaluation of the usage of imprecise operators inside a VBSME. The synthesis results are presented in Section V, and comparisons with related works are presented in Section VI. Finally, some conclusions are drawn in Section VII.

II. Lower-Part-OR Adder

Arithmetic operators and circuits are essential in any digital system and can significantly influence the achievable overall performance [13]. In fact, due to the carry propagation principle, arithmetic operators are not only the main responsible for the delay but are also the cause of most of the power dissipation in most digital circuits [14]. To solve this fundamental problem, approaches of approximate computing are commonly used, such as shortening or truncating the carry chain, thereby introducing some level of imprecision in the results.

Many approximate operators have been proposed in the literature. Some examples are Almost Correct Adder [15], Lower-Part-OR Adder [16], Error-Tolerant Adder [17], Accuracy-Configurable Adder [18], Generic Accuracy Configurable Adder [19], among others. The work herein presented considers the usage of Lower-Part-OR Adders (LOA) in video encoding, by replacing the Ripple-Carry Adders (RCA) in the SAD units of a VBSME, with the main objective of decreasing the energy consumption while minimizing the impact in the coding efficiency.

LOA structures an addition into two smaller sections. The upper-section (most significant bits) performs the regular precise addition. For the least significant bits (lower-part) there is a simplification, the carry chain propagation is eliminated, as depicted in Fig. 1.

While well known structures can be used to design the precise adder, such as Ripple-Carry Adders or Carry Look-Ahead Adders, in the lower part bitwise OR is applied to the inputs and no carry is generated. To generate a carry-in for the upper part, an extra AND gate is used in the most significant bits of the imprecise part, with the goal of decreasing imprecision [16].

III. Energy-Efficient VBSME Architectures

The VBSME architectural exploration has used as reference a VBSME architecture [21], previously developed for real-time processing of high definition videos. Although this previously designed version of VBSME targeted the implementation of the H.264/AVC standard, the proposed technique also works for the HEVC standard, due to its strategy of reusing SAD values of smaller block sizes. By using this architecture as a reference, new versions were designed using three types of LOA operators, with 3-, 4- and 5-bit in the imprecise part. These imprecision levels were defined for 8-bit width operators, which is adopted in this work. Thus, it was possible to evaluate the imprecision for half width of the operator (4-bit), and in both directions of imprecision, by increasing (5-bit) or decreasing (3-bit) it. These new versions of VBSME architectures were called energy-efficient VBSME (E-VBSME) in this work.

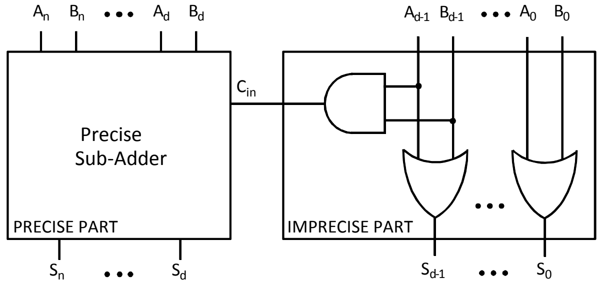

Fig. 1. Lower-Part-OR Adder Structure.

The VBSME architecture operates over 16x16 blocks, by merging the SAD (Sum of Absolute Differences) values of smaller (4x4) blocks to calculate SAD values of larger blocks. Hence, the VBSME architecture uses a 4x4 ME module as its basic structure, as depicted in Fig. 2, and adders to group the 4x4 SADs forming the SADs of larger blocks. The energy efficiency exploration strategy considers the replacement of the original Ripple Carry Adders (RCA) by the three options of LOA operators previously discussed. However, such replacement is only applied in the first step of the SAD calculations (subtraction), in order to avoid an excessive accumulated imprecision. Over a 16x16 pixels search area there are 13 candidate blocks in a row and 13 candidate blocks in a column for a 4x4 block. Thus, 169 candidate blocks are compared with the current block, one at a time, in order to find the best match. In this way, in each level of the architecture there are 169 SAD values, each one corresponding to one candidate block.

The block diagram of the 4x4 ME architecture shown in Fig.2 is the most important architectural module of the proposed structure, the other VBSME architectural modules are explained in detail in [21]. Some signals have been hidden in Fig. 2 to allow a better viewing, particularly the control ones. The main 4x4 ME modules are the SAD rows and the corresponding Processing Cores (PCs), comparators, memories and the memory manager.

The PC (see Fig. 2) calculates the SAD between a row of the current block and a row of the candidate block. The SAD row is formed by four PCs, being the first three PCs responsible for processing four candidate blocks each while the last one is responsible for processing just one candidate block. So, the SAD row processes the 13 candidate blocks within a row. The first SAD row calculates the SAD values for the 13 candidate blocks that begin at the first row of the search area, comparing the current block to these candidate blocks. The second SAD row does the same process for the 13 candidate blocks at the second row of the reference area. This process is carried out until, in the thirteenth SAD row, the current block is compared with the 13 last candidate blocks. In this way, the current block is compared to the 169 candidate blocks to find the best match.

The SAD calculation architecture was hierarchically designed with 13 rows, each one with 4 PCs, which means that the 4x4 ME contains 52 PCs (13 rows with four PCs per row).

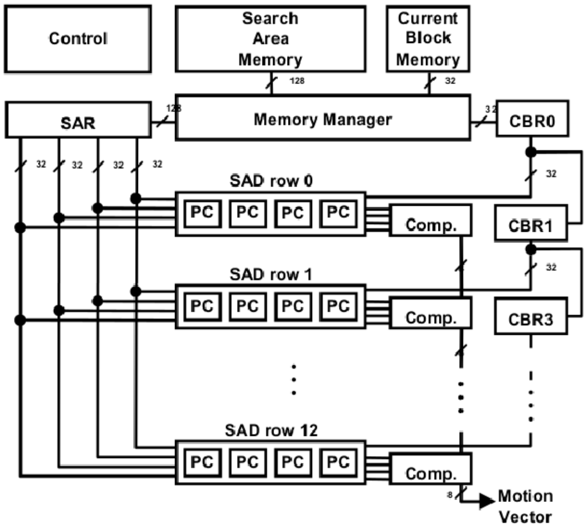

Fig. 2. Block diagram of the VSBME 4x4 ME module [21].

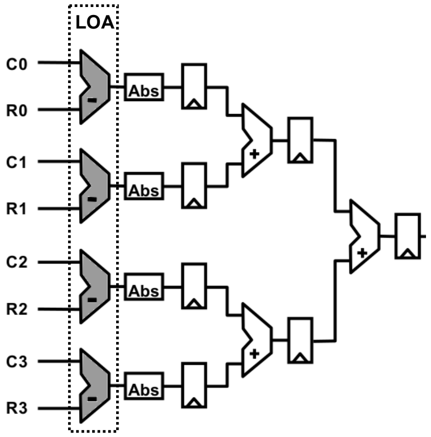

Fig. 3. PC architecture with the LOA operators highlighted.

Fig. 3 shows the PC architecture. Each PC receives four samples from the current block (C0 to C3) and four samples from the candidate block (R0 the R3). The SAD that is computed by a PC is a partial value (SAD of a row) and needs to be added to the SAD values from other rows to generate the total SAD of a candidate block. Each SAD row groups four PCs and generates the total SAD of each candidate block in this row.

It is important to notice that there are 364 arithmetic operators in each SAD row. Among these operators, 208 are subtractors at the first PC stage. These subtractors are highlighted in Fig. 3 and they are the target of this architectural exploration: they were substituted by LOA operators. The operators in the next pipeline stages (accumulation) were not substituted, otherwise the error generated in the first stage will be accumulated with the errors generated in the next stages.

Hence, by considering the diagram in Fig. 1, the closer to *n* is the value of *d*, the greater is the reduction of area, delay and power. Conversely, the lower is the accuracy. At this respect, LOA provides the smallest power and area metrics. However, it has the highest approximation errors among the considered approximate operators [14]. Nevertheless, since LOA is applied in this work to individual small wide (8-bit) operands, and even smaller bit width in the imprecise part, these errors are not a problem.

Hence, at the end of this design exploration exercise, three new VBSME architectural versions were obtained, by considering the three imprecision levels previously referred. Then, it was conducted a thorough evaluation of the coding efficiency impacts caused by this design strategy, through an extensive software evaluation, that is presented in the next section.

IV. Impact of Computing Imprecision on Coding Efficiency

To evaluate the impact of the proposed computing imprecision on the resulting coding efficiency, the considered approximate operators were described in C++ to replace part of the source code in the HM 16.12 [12] HEVC reference software. The three considered configurations with different levels of imprecision (3-, 4-, and 5-bit imprecision), as well as the unmodified version without imprecision (original HM 16.12 version) were tested and compared. All these coder configurations were defined to guarantee that the executed software has the same behavior than the E-VBSME hardware architecture presented in the previous section, allowing a fair and precise evaluation of the impacts of the introduced imprecision. For such purpose, the LOA operators were inserted in the HM, only into the first stage of the SAD operation, as it was defined in the previous section. This first operation corresponds to the subtraction needed to generate the absolute differences of SAD.

The results of this evaluation were obtained by encoding the twenty test video sequences recommended in the Common Test Conditions (CTC) [20]. The test sequences are classified in five classes: B (1080p), C (WVGA), D (WQVGA), E (720p), and F (Screen Content). The configuration of this experiment corresponds to the Low Delay P Main, with the four QP values (22, 27, 32, and 37) also recommended in the CTC. Table I summarizes the results of this evaluation. The presented Bjøntegaard Delta rate (BD-Rate) values in Table I have been calculated for the four different QP parametrizations. According to the obtained values, the impact varies following to the characteristics of each video class but the effect can be considered negligible.

Fig. 4 depicts the expected behavior: the greater is the imprecision level, the greater is the coding efficiency degradation. Considering a 3-bit imprecision setup, the results show, on average, an increase of only 0.6% in BD-Rate for the luminance and 0.45% for chrominance. By comparing the results obtained for 4-bit imprecision with the 3-bit imprecision, it is observed that the impact on the BD-Rate only increases by 0.6% for luminance and 0.35% for chroma components. In the case where 5-bit imprecision is used the losses were more representative, but still with a magnitude of only 2.5% for luminance and 1.85% for chrominance.

TABLE I. BD-RATE DEGRADATION (%) FOR 3 LEVELS OF IMPRECISION.

Class	3-Bit Imprecision			4-Bit Imprecision			5-Bit Imprecision		
	Y	U	V	Y	U	V	Y	U	V
B (1080p)	0.7	0.4	0.2	1.3	0.8	0.8	2.5	1.5	1.4
C (WVGA)	0.6	0.3	0.3	1.3	0.9	0.9	2.5	1.8	1.8
D (WQVGA)	0.7	0.6	0.3	1.5	1.3	0.8	3.0	2.2	2.2
E (720p)	0.4	0.0	0.0	1.2	0.4	0.6	2.3	1.2	1.3
F (S. C.)	0.8	1.1	1.1	0.5	0.6	0.8	2.1	2.1	2.6
Average	0.6	0.5	0.4	1.2	0.8	0.8	2.5	1.8	1.9

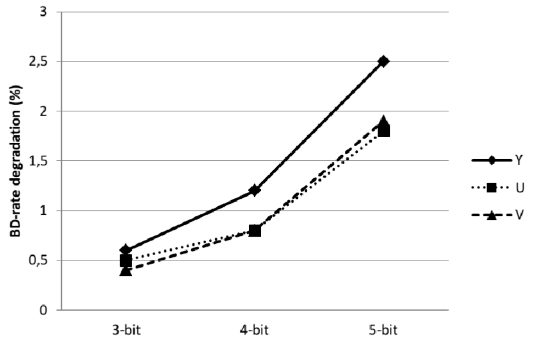

Fig. 4. BD-rate degradation (%) for 3 levels of LOA imprecision.

In the whole, it can be concluded that the coding efficiency losses using LOA depend on the level of imprecision, which is introduced by this type of operators, but are highly acceptable for all analyzed cases. The power synthesis results, including the power analysis, are presented in the next section.

V. HARDWARE IMPLEMENTATION AND EVALAUATION

The proposed VBSME architecture was hierarchically described in VHDL in four different versions, the original and the three versions using LOA. The architectures were synthesized using a 45nm@1.1V Nangate standard cell library. Cadence Encounter RTL Compiler tool was used for the syntheses, configured to high effort for power, synthesis and mapping.

The syntheses of the three E-VBSME versions focused on real-time video in HD720p (1280 x 720) and HD1080p (1920 x 1080) resolutions at 30 and 60 frames per second.

Firstly, only the PC, presented in Fig. 3, was evaluated. The PC contains the SAD calculations, which is the most important VSBME module. Table II presents the evaluation of the four PC versions, which refer to the operators used in the first stage of SAD calculation. RCA version used Ripple-Carry Adders. The versions named LOA refer to the usage of Lower-Part-OR Adders, and the number following the abbreviation indicates the bit width of the imprecise part (e.g., LOA3 refers to a LOA operator with 3-bit imprecision). The operating frequencies used to obtain the power results are also presented in Table II. These frequencies were defined to allow the E-VBSME hardware to operate in real time at different resolutions and frame rates. Since the VSBME architecture has 53 PC instances, one of these PCs was selected to be evaluated and the results are presented in Table II.

TABLE II. POWER RESULTS FOR THE VARIOUS PC CONFIGURATIONS.

Resolution	Freq. (MHz)	Power (mW)			
		RCA	LOA3	LOA4	LOA5
HD1080p@60fps	497.66	0.307	0.271	0.256	0.243
HD1080p@30fps	248.83	0.185	0.167	0.156	0.147
HD720p@60fps	221.18	0.172	0.148	0.142	0.134
HD720p@30fps	110.59	0.102	0.089	0.084	0.081

As expected, the power dissipation decreases with the imprecision increase. In Table II, among the power results, the highest were obtained for the RCA and the smallest ones to the LOA5 version. Accordingly, the reached power gains with LOA vary from 9.7% to 22.1%, 16.6% in average, when compared with the RCA version. These are expressive gains, considering the insignificant decrease of the coder efficiency presented in previous section (3% in the worst case).

Table III presents the area results for the same four PC versions. By analyzing these results, one can conclude that the usage of LOA did not significantly impacts in terms use of hardware resources. Considering the chip area, the use of LOA caused a reduction between 2% and 5% of the total circuit area. When considering the gate count, the use of LOA increases the number of used gates from 5.5% to 7% when compared to those used by RCAs. These results can be explained through the use, by the synthesis tool, of complex gates specialized in ripple carry adder operations, which are available at the used standard-cells library. Then, with larger cells, the RCA version will use a larger chip area even using fewer gates than the LOA versions.

The second evaluation considered the complete VSBME architecture, where the PCs are inserted. This evaluation considered the same scenario previously described, targeting the same operation frequencies and using the same stimulus. The power results are presented in Table IV. Since the VSBME has other modules that do not use LOA adders, the power gains are a bit smaller than those presented in PCs, but these gains are still important. The gains vary from 0.5mW (LOA3 @ 110.59MHz) to 2.46mW (LOA5 @ 497.66MHz), when compared with RCA.

TABLE III. PC ARCHITECTURE AREA RESULTS.

	RCA	LOA3	LOA4	LOA5
Area (μm^2)	1,525	1,495	1,480	1,448
Gate Count (Kgates)	273	288	292	289

TABLE IV. POWER RESULTS FOR THE VSBME ARCHITETURES.

Resolution	Freq. (MHz)	Power (mW)			
		RCA	LOA3	LOA4	LOA5
HD1080p@60fps	497.66	21.40	19.66	18.94	18.43
HD1080p@30fps	248.83	12.84	11.81	11.40	11.11
HD720p@60fps	221.18	11.72	10.83	10.52	10.25
HD720p@30fps	110.59	7.32	6.81	6.64	6.51

By analyzing the results of Table IV it is possible to obtain the percentage of power savings provided by LOA operators. LOA5 reached the highest percentage of power savings, 11.5% when running at 497.66MHz. As expected, power savings increase as the level of imprecision increases. Summarizing, the obtained power savings vary between 7% (LOA3 @ 110.59MHz) and 11.5% (LOA5 @ 497.66MHz). These gains are highly significant when compared with the negligible decrease of the encoding efficiency (3% in the worst case).

The complete VSBME architecture using LOA synthesis results also showed small variations in terms of hardware resources usage. Table V shows the area and gate count results for the different versions of E-VBSME. As it is shown in Table V, the VSBME implementation results follow the same behavior of the PC results. The total area decreased with the imprecision increase, with gains between 0.1% and 1.5% when compared with the RCA version. The opposite behavior was found in the gate count results, with losses between 3.6% and 4.3%. Again, using adder specialized complex gates, the synthesis tool uses gates with larger sizes, leading to larger RCA chip area even when using less gates than in the LOA versions.

A fairer way to compare the implemented versions of E-VBSME is to analyze the relation between the obtained power savings and the resulting BD-Rate, in order to evaluate the general efficiency of the proposed solutions. In this way, it is possible to measure how much coding efficiency is lost to allow the achieved energy consumption reduction. The results are presented in Table VI, the higher the value, the better is the result. By considering this relation, the best setup was the LOA3, since it presents the highest values of efficiency for all evaluated operation frequencies.

TABLE V. VSBME ARCHITETURES AREA RESULTS.

	RCA	**LOA3**	**LOA4**	**LOA5**
Area (µm^2)	179,6	179,4	178,6	177,0
Gate Count (Kgates)	29,8	30,9	31,1	31,0

TABLE VI. EFFICIENCY (POWER SAVINGS/BD-RATE)

Resolution	**LOA3**	**LOA4**	**LOA5**
HD1080p@60fps	2.5	1.9	1.2
HD1080p@30fps	1.5	1.1	0.7
HD720p@60fps	2.2	1.0	0.6
HD720p@30fps	1.3	0.6	0.4

VI. COMPARISON WITH RELATED WORKS

Several ME architectures have been published in the literature, such as [7], [8], [9], [10], and [11]. Unfortunately, it is difficult to make a fair comparison of all these architectures, including the one presented in this work, since these published works were developed by targeting different coding standards, synthesized in different technologies, focused on different resolutions, and implemented with different configurations (search area, block sizes, support to fractional prediction, among others).

Table VII presents a summary of some of these published solutions, identifying key features of these works, such as: targeted video standard; supported resolution, tools, and block sizes; search range; search algorithm; CMOS technology; operating frequency; number of gates; reached throughput and power. The resolution is presented with the related frame rate (the number after the "@" in Table VII).

Even with important structural differences, it is possible to conclude that the proposed approaches with LOA present the best power results among all the solutions presented in Table VII, even when running the highest operating frequency. Unfortunately, only a few recent published works present power results. When compared with two of these works, the LOA setups reached, in the worst case (LOA3), a power consumption that is 5.1 times lower for a throughput 1.8 times lower when compared with [7], and a power 21.5 times lower for a 3.3 lower throughput when compared with [8].

The LOA versions also present the best results in terms of used gates among all compared works. Part of these differences result from the distinct reported supported tools, algorithms, block sizes and project options. The worst result, in terms of gate count, was achieved with LOA4, with 31.1Kgates. Even so, this LOA version used 11.3 times less hardware than [7], with a throughput only 1.9 lower. When compared with [8], LOA4 uses 5.1 times less gates, reaching a throughput 3.3 times lower. The LOA4 also uses 58.8 less hardware than [9], with a throughput only 2 times lower. Finally, LOA4 requires 25 less gates than [10], reaching a throughput 4 times lower.

VII. CONCLUSIONS

This manuscript presented three versions of an energy-efficient variable block size motion estimation architecture (E-VBSME) using approximate operators. It proposed the usage of LOA adders to perform the SAD calculations, in order to reduce energy consumption. Only a reduced number of operators were substituted, intending to restrict the coding efficiency losses. It is shown that this approach presents negligible impacts in the coding efficiency, when evaluated using the HEVC reference software.

Three levels of imprecision were evaluated, with BD-Rate increasing from 0.6% to 2.5%. The three versions of the E-VBSME architecture were synthesized for Nangate 45nm standard cells technology. Extensive comparisons were done between the original version (using RCAs only) and the three versions with LOA. Synthesis results indicate power savings from 9.7% to 22.1%, when considering only the Processing Core (where the SAD calculations are done). The global E-VSBME architecture reached 7% to 11.5% power savings when compared with the original version of the architecture.

The comparison with related works showed rather competitive results. When compared with most relevant works of the state of the art, the E-VBSME reached the best power and area results, and the best relation between power versus throughput. These results showed that the use of imprecision in video coding applications is a suitable alternative to handle with strict power restrictions.

TABLE VII. COMPARISONS WITH RELATED WORKS.

Related Works	Li [7]	Cao [8]	Sinangil [9]	Jou [10]	Porto [21]	LOA3	LOA4	LOA5
Standard	H.264/AVC	H.264/AVC	HEVC	HEVC	H.264/AVC	H.264/AVC, HEVC	H.264/AVC, HEVC	H.264/AVC, HEVC
Algorithm	LBA and PDA	FS	TZS	PEPZS	FS	FS	FS	FS
Search Range	16x16	33x33	64x64	64x64	16x16	16x16	16x16	16x16
Block Size	4x4, 4x8, 8x4, 8x8, 8x16, 16x8, 16x16	4x4, 4x8, 8x4, 8x8, 8x16, 16x8, 16x16	16x16, 32x32, 64x64	4x8, 8x4, 8x8, 8x16,16x8,16x16, 32x32, 64x64	4x4, 4x8, 8x4, 8x8, 8x16, 16x8, 16x16	4x4, 4x8, 8x4, 8x8, 8x16, 16x8, 16x16	4x4, 4x8, 8x4, 8x8, 8x16, 16x8, 16xe16	4x4, 4x8, 8x4, 8x8, 8x16, 16x8, 16x16
Supported Tools	IME	IME	IME, FME	IME, FME	IME	IME	IME	IME
Resolution	1280x720@25	1280x720@45	3840x2160@30	4096x2048@60	1920x1080@60	1920x1080@60	1920x1080@60	1920x1080@60
Technology	0.18 um	0.18 um	65 nm	90 nm	45 nm	45 nm	45 nm	45 nm
Frequency (MHz)	150	180	200	270	497.66	497.66	497.66	497.66
Gates (Kgates)	350	160	1830	778.7	29.8	30.9	31.1	31.0
Throughput (Mpixel/sec)	230.4	414.7	248.8	503.3	124.4	124.4	124.4	124.4
Power (mW)	101.5	423	n/a	n/a	21.40	19.66	18.94	18.43

Hence, the obtained results with the LOA operators drive further investigations, through the adoption of new and dedicated imprecise operators targeting the specificities of video coding. Given the obtained results, the usage of approximate operators in other video coding modules also deserves careful attention.

ACKNOWLEDGMENT

The authors would like to acknowledge Federal University of Pelotas (UFPel), in Brazil, and Institute for Systems Engineering and Computers (INESC), in Portugal, where this work was developed. The authors also have a special acknowledgment to CNPq, CAPES and FAPERGS to support this work. This work was partially supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) under project number UID/CEC/50021/2013.

REFERENCES

[1] G. Sullivan, J. Ohm, W. Han. "Overview of the high efficiency video coding (HEVC) standard", in IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649-1668, 2012.

[2] J. Han, M. Orshansky. "Approximate computing: an emerging paradigm for energy-efficient design", in IEEE European Test Symposium, pp. 1-6, 2013.

[3] V. Chippa, S. Venkataramani, S. Chakradhar, K. Roy, A. Raghunathan. "Approximate computing: an integrated hardware approach", in Asilomar Conference on Signals, Systems and Computers, pp. 111-117, 2013.

[4] V. Gupta, D. Mohapatra, S. Park, A. Raghunathan, K. Roy. "Impact: imprecise adders for low-power approximate computing", in IEEE International Symposium on Low Power Electronics and Design, pp. 409-414, 2011.

[5] A. Raha, H. Jayakumar, V. Raghunathan. "A power efficient video encoder using reconfigurable approximate arithmetic units", in IEEE International Conference on VLSI Design and International Conference on Embedded Systems, pp. 324-329, 2014.

[6] X. Gao, W. Lu, D. Tao, X. Li. "Image quality assessment and human visual system", in SPIE Video Communications and Imagem Processing, vol. 7744, pp. 77440Z-1-77440Z-10, 2010.

[7] P. Li, H. Tang, "A low-power VLSI implementation for variable block size motion estimation in H.264/AVC", in IEEE International Symposium on Circuits and Systems, pp. 2972-2975, 2010.

[8] W. Cao, H. Hou, J. Tong, J. Lai, H. Min, "A High-performance reconfigurable VLSI architecture for VBSME in H.264", in IEEE Transactions of Consumer Electronics, vol. 54, no. 3, 1338-1345, 2008.

[9] M. E. Sinangil, V. Sze, Z. Minhua, A. P. Chandrakasan, "Cost and coding efficient motion estimation design considerations for High Efficiency Video Coding (HEVC) standard," in IEEE Journal of Selected Topics in Signal Processing, vol. 7, no. 6, pp. 1017-1028, 2013.

[10] S-Y. Jou, S-J. Chang, T-S. Chang. "Fast Motion Estimation Algorithm and Design for Real Time QFHD High Efficiency Video Coding", in IEEE Transactions on Circuits and Systems for Video Technology, vol. 25, no. 9, pp. 1533-1544, 2015.

[11] P. Nalluri, L. Alves; A. Navarro. "High speed SAD architectures for variable block size motion estimation in HEVC video coding", in IEEE International Conference on Image Processing, pp 1233-1237, 2014.

[12] HEVC Reference Software (HM) Repository. <https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/>, 2016.

[13] F. Frustaci, M. Lanuzza, P. Zicari, S. Perri, P. Corsonello. "Designing high-speed adders in power-constrained environments", in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 56, no. 2, pp. 172-176, 2009.

[14] S. Dutt, S. Nandi, G. Trivedi. "A comparative survey of approximate adders", in IEEE International Conference Radioelektronika, pp. 61-65, 2016.

[15] A. Verma, P. Brisk, P. Ienne. "Variable latency speculative addition: a new paradigm for arithmetic circuit design", in IEEE Design, Automation and Test in Europe Conference and Exhibition, pp. 1250-1255, 2008.

[16] H. Mahdiani, A. Ahmadi, M. Fakhraie, C. Lucas. "Bio-inspired imprecise computacional blocks for efficient vlsi implementation of soft-computing applications", in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 57, no. 4, pp. 850-862, 2010.

[17] N. Zhu, W. Goh, G. Wang, K. Yeo. "Enhanced low-power high-speed adder for error-tolerant application", in IEEE International SOC Design Conference, pp. 323-327, 2010.

[18] A. Kahng, S. Kang. "Accuracy-configurable adder for approximate arithmetic designs", in ACM/EDAC/IEEE Design Automation Conference, pp. 820-825, 2012.

[19] M. Shafique, W. Ahmad, R. Hafiz, J. Henkel. "A low latency generic accuracy configurable adder", in ACM/EDAC/IEEE Design Automation Conference, pp. 1-6, 2015.

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

[21] R. Porto, L. Agostini, S. Bampi. "Hardware design of the H.264/AVC variable block size motion estimation for real-time 1080HD video encoding", in IEEE Computer Society Annual Symposium on VLSI, pp. 115-120, 2009.