Journal of Real-Time Image Processing (2020) 17:1685–1701
https://doi.org/10.1007/s11554-019-00934-2

ORIGINAL RESEARCH PAPER

UHD 8K energy-quality scalable HEVC intra-prediction SAD unit hardware using optimized and configurable imprecise adders

Roger Porto[1,2] · Marcel Correa[1,2] · Jones Goebel[1] · Bruno Zatt[1] · Nuno Roma[3,4] · Luciano Agostini[1] · Marcelo Porto[1]

Received: 30 April 2019 / Accepted: 28 November 2019 / Published online: 11 December 2019

Abstract
Real-time digital video coding became a mandatory feature in current consumer electronic devices due to the popularization of video applications. However, efficiently encoding videos is an extremely processing/energy-demanding task, especially at high resolutions and frame rates. Thus, the limited energy resources and the dynamically varying system status (such as workload, battery level, user settings, etc.) require energy-efficient solutions capable to support run-time energy-quality scalability. In this work, we present an energy-quality scalable SAD Unit hardware architecture for the HEVC intra-frame prediction targeting real-time processing of UHD 8K (7680×4320) videos at 60 frames per second. Approximate computing is used to provide energy-quality scalability by employing configurable imprecise operators. The proposed Energy-Quality scalable architecture supports four operation points: precise computing, and 3-bit, 5-bit or 7-bit imprecision. When implemented in a 45-nm technology using Nangate standard cells library and running at 269 MHz, the proposed architecture consumes from 8.42 to 7.38 mJ to process each UHD 8K frame, according to the selected imprecision level. As a drawback, the coding efficiency (measured in BD rate) is reduced from 0.28 to 1.72%. Compared to the related works, this is the only intra-frame prediction SAD unit able to provide energy-quality scalability.

Keywords Energy-quality scalability · Video coding · Intra-prediction · SAD · Scalable hardware design · Approximate computing

1 Introduction

The omnipresence of digital videos and the increasing demand for higher resolutions (Full HD, UHD 4K, and UHD 8K), higher frame rates (60 fps, 120 fps, etc.), better color representations (HDR—high dynamic range), and immersive experience (3D and omnidirectional videos) drastically increased the amount of video content to be processed, stored, and transmitted. As a result, video traffic over the internet consumed more than 56 exabytes per month in 2017, using 75% of the global internet traffic [1]. In this trend, it is expected that video contents will consume 240 exabytes per month by 2022, or 82% of the total internet traffic [1]. Consequently, the pressure between the fast-increasing traffic and the limited network expansion has been pushing the evolution of video encoders along the last couple of decades. As a response, each new video coding standard generation has introduced novel/improved algorithms and data structures, to improve the coding efficiency.

An efficient implementation of a video encoder represents a large challenge when it comes to real-time systems, due to the huge computational effort that is demanded. For instance, the state-of-the-art video coding standard—HEVC (high efficiency video coding) [2]—demands a coding effort up to five times higher than its predecessor H.264, to provide twice the coding efficiency [3, 4]. However, the design of encoding systems becomes even more challenging as most video-capable devices are mobile systems featuring limited energy resources/battery capacity. These devices also must be able to capture digital videos, requiring efficient video encoder implementations to store or transmit the captured

✉ Roger Porto
recporto@inf.ufpel.edu.br

[1] Video Technology Research Group, Group of Architectures and Integrated Circuits, Federal University of Pelotas (UFPel), Pelotas, RS 96010-900, Brazil

[2] Sul-Rio-Grandense Federal Institute of Science and Technology (IFSul), Bagé, Brazil

[3] Instituto Superior Técnico (IST), Universidade de Lisboa, Lisbon, Portugal

[4] INESC-ID, Lisbon, Portugal

videos. According to [1], about 10 exabytes of internet traffic were generated per month from/to mobile devices in 2017, and this number may exceed the 68 exabytes by 2022. Therefore, there is a prevailing need for energy-efficient encoding solutions able to sustain high coding efficiency and long battery life. In this scenario, the employment of hardware acceleration has become a mandatory approach to deal with the severe performance and energy constraints.

A variety of low-power/energy-efficient solutions for several video encoder functional units have been proposed in the literature, supported on several coding standards such as HEVC, H.264, VP-9/10, etc. These proposals include hardware architectures for intra- and inter-prediction [5–10], transforms and quantization [11–13], filters [14–16], and entropy encoding modules [17–19]. The employed techniques to reduce power/energy include algorithmic simplification [20, 21], data subsampling [22, 23], and approximate computing [24–26], among others. However, these solutions implement power/energy-oriented optimizations that pose quality losses. In the scope of this work, quality refers to the application quality—instead of video quality—defined as the coding efficiency calculated by a function of bit rate and objective video quality [27]. Such losses are acceptable, especially for real-time systems, as video processing is known as an error-tolerant application [28], i.e., resilient to numerically imprecise partial results. However, defining the optimal balance between energy consumption and quality is not a simple task, since it highly depends on the video content (resolution, frame rate, motion, texture, etc.). Additionally, user preferences and ever-changing system status (battery status, workload, etc.) may modify the desired Energy-Quality (EQ) tradeoff. Thus, it is necessary to develop efficient and effective EQ-scalable video coding systems that support run-time adaptation, by navigating through the distinct EQ tradeoff operation points.

Approximate computing arises as a major approach to reduce energy consumption, providing an additional knob to control EQ scalability. Some hardware architectures employing approximate computing for the HEVC encoder have already been proposed for the motion estimation [25, 26] and for the transforms unit [24, 29]. However, none of these solutions provide support to EQ scalability. Moreover, no approximate solution targeting the intra-frame prediction unit has been found in the literature, leaving an important research gap. Intra-frame prediction is a critical task at the encoder side, being responsible for reducing intra-frame redundancies, by selecting the best intra-prediction mode out of 35 possible modes and five block sizes: 4×4, 8×8, 16×16, 32×32 and 64×64 [30] (see Sect. 2). This is particularly relevant when considering that HEVC has a much larger exploration space when compared to previous standards, such as H.264, that only defines nine prediction modes and three block sizes [31]. However, evaluating multiple prediction modes requires multiple calculations of the distortion criterion, becoming one of the major processing/energy bottlenecks within the prediction process.

The SAD (sum of absolute differences) [32] is the most used criterion in real-time systems and represents a major point for optimization. As an example, the encoding of the NebutaFestival sequence [33] (2560×1600 resolution) by the HEVC reference software (HM 16.2) [34] using the all intra configuration [33] requires (on average) almost 12 million SAD calculations per frame. Considering all the supported block sizes, a total of 716.8 million samples are compared per frame using SAD. For a 2-h video, a total of 309 trillion samples are compared. Therefore, proposing an efficient and scalable intra-prediction solution employing approximate computing to optimize the SAD operators is a highly promising approach and it will be the focus of this work. This claim is further supported by our own experiments (considering CTC Class A video sequences [33] in HM 16.2 encoder [34]—see Sect. 3.2 for methodology details) which show that 49.92% and 22.37% of the encoding time are dedicated to inter-prediction and intra-prediction, respectively, i.e., the two prediction modules consume 73.29% of the total encoding time. Hence, since the SAD [or the Sum of Absolute Transformed Differences (SATD)] is required to evaluate all possible prediction candidates, optimizing the SAD calculation is a key aspect to improve the power efficiency of the global encoder.

By considering this observation, an energy-quality scalable hardware architecture of a massively parallel SAD calculation unit targeting the HEVC intra-prediction module and featuring an arithmetic operator with multiple levels of imprecision will be presented. The proposed architecture is a considerable enhancement of the computing unit proposed in [31], which was able to process UHD 8K videos in real time but did not offer any configurability. Moreover, the newly presented SAD solution can be also used in inter-prediction architectures or as a basic block for other similarity criteria, including the SATD.

The main contributions are described below:

- Evaluation of imprecise adder operators: six approximate adder operators were evaluated in the context of the HEVC intra-prediction considering power, quality, delay, area, and power–delay product;
- Definition of viable EQ operation points: supported on the power characterization of the operators, different LOA (Lower-Part-OR Adder) implementations considering distinct number of approximate bits were used to define four EQ operation points;
- Design of an optimized and configurable adder: an optimized operator was designed, supporting run-time selection among four EQ operation points with reduced area overhead;

- Conception of an EQ-scalable intra-frame SAD unit: a new HW architecture to implement the SAD unit and to provide real-time performance for up to UHD 8K at 60 fps was designed. The proposed architecture features 35 SAD trees and allows run-time EQ scalability by selecting among four EQ operation points.

The remainder of the paper is as follows. The next section briefly reviews the HEVC intra-prediction module definition. Section 3 presents the considered set of imprecise adders, as well as a preliminary evaluation of these adders by considering the coding efficiency and hardware implementation. These experiments were used to define which imprecise adder is the most appropriate to design an energy-quality scalable architecture targeting the SAD calculation. Section 4 presents the conducted experiments to define the most convenient operation points of the intra-prediction SAD unit and Sect. 5 presents the proposed energy-quality scalable SAD unit architecture. Section 6 discusses the reached results and compares them with related works. Finally, conclusions are addressed and presented in Sect. 7.

2 HEVC intra-prediction

The HEVC intra-prediction module supports 35 prediction modes (33 directional and two non-directional modes) [30]. The directional modes are suitable for areas with directional structures and the remaining two modes, planar and DC, are suitable for homogeneous areas.

Figure 1 depicts an 8 × 8 example block (white squares), to be predicted using 33 previously encoded reference samples (non-white squares). Generically, $4N+1$ reference samples are needed to predict each $N \times N$ block. Every intra-predicted block must go through three different steps: pre-filtering of reference samples, sample prediction, and post-filtering of predicted samples [30]. The pre-filtering is used when adjacent reference samples have notable discrepancies in their values. In these cases, unwanted artifacts may appear in blocks predicted by some combinations of block size and prediction mode. To mitigate this effect, smoothing filters are applied to the reference samples before block prediction. The adopted filter is a function of the block size and the used prediction mode.

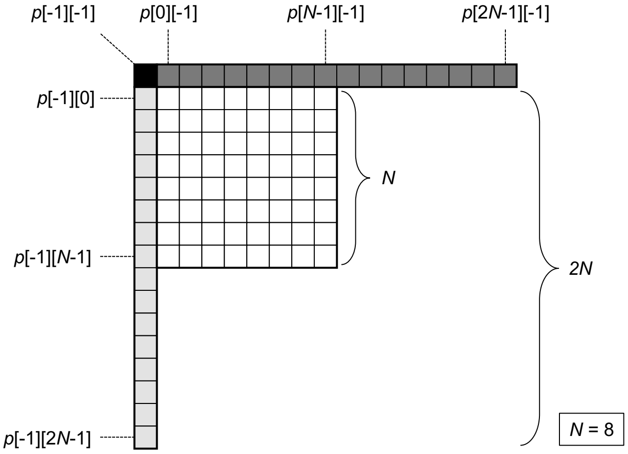


Fig. 1 An 8 × 8 block to be predicted (white squares) using 33 reference samples (non-white squares)

The sample prediction step is where the prediction actually occurs, i.e., where the prediction blocks are computed using the 35 available prediction modes and it is where these blocks are compared with the current block to select which blocks are the best options to encode the current block. This step is the core of the intra-prediction operation.

Considering a 64×64 Coding Tree Block (CTB) [35], the predictors are applied over four different block sizes: 4×4, 8×8, 16×16 and 32×32 [30]. Since there are 35 prediction modes, 140 combinations of prediction parameters are allowed. The considered predictions are compared to the original block using some distortion criterion [32]. The HM [34] implementation of HEVC intra-prediction allows the use of the SAD and SATD [32] distortion criteria. The distortion criterion that shall be considered in this article is the SAD, since it is the most frequently used in video encoding [32]. This distortion must be calculated for all available block sizes inside a CTB, from 4×4 to 64×64 [36]. Since there are no predictors for 64×64 blocks, when the four 32×32 blocks that form a 64×64 block use the same prediction, these blocks are joined together to generate the prediction for the 64×64 block [34].

The last step of the intra-prediction module is the post-processing filter, which is used to reduce the discontinuities that some of the intra-prediction modes can generate for the predicted samples located at the top and left borders of the predicted block [30].

Since the HEVC intra-prediction supports four block sizes and 35 prediction modes, the evaluation of all these prediction candidates through the HEVC rate-distortion optimization (RDO) process [35, 36] is impractical. As a result, the HEVC reference software [34] uses two heuristics to define some local decisions intending to reduce the global encoder complexity. The first one is the rough mode decision (RMD) [37], which selects only a few number of prediction modes to be evaluated by the full RDO: eight for 4×4 and 8×8 blocks and three for 16×16 and 32×32 blocks [37]. The second heuristic is used to increase the coding efficiency, by adding three additional most probable modes (MPMs) [30] in the RDO evaluation, for each block size.

3 Imprecise adder structures

The main idea that is explored in this article is the use of distinct levels of imprecise arithmetic to scale the power consumption of a high-throughput intra-prediction architecture. In particular, since the sum of absolute differences (SAD) is the dominant operator in the intra-prediction implementation, it was selected to use imprecise adders. The first investigation to support this design was the selection of the most adequate imprecise operator that reaches the best results in this scenario. This selection was based on a thorough evaluation of these operators when used in the HEVC intra-prediction.

3.1 Imprecise arithmetic operators

There are many imprecise operators in the literature, and this article focuses on six of them: Accuracy-Configurable Adder [38], Carry Cut-Back Adder [39], two versions of the Error-Tolerant Adder, [40] and [41], Generic Accuracy-Configurable Adder [42], and Lower-Part-OR Adder [43].

The Accuracy-Configurable Adder (ACA-II) was proposed in [38]. It segments the addition, distributing the imprecision through the used sub-adders. Three overlapped sub-adders are used to reduce the carry propagation.

The Carry Cut-Back Adder (CCB), proposed in [39], is also a segment-based approximate operator. It uses the carry propagate signal from the most significant bits (MSB) to cut the carry propagation of low significance bits (LSB). CCB uses manifold propagate signals and multiplexers to shorten the propagation chain, reducing the adder critical path [39].

The Error-Tolerant Adders (ETA) were proposed in [40, 41] and two versions of this adder are considered in this article: ETA-I and ETA-IV. The ETA-I [40] is an approximate adder that splits the addition into two non-overlapped sub-adders. The imprecision is only applied in the LSB. The imprecise ETA-I sub-adder checks every bit position from left to right (MSB to LSB). If both input bits are “0” or different, normal one-bit addition is performed and the operation proceeds to next bit position. Otherwise, if both input bits are “1”, the checking process stopped and from this bit onward, all sum bits to the right are set to “1” [40]. The ETA-IV [41] also splits the addition into two non-overlapping sub-adders, but in this case the carry propagation is reduced through specialized units that generate the carries from the imprecise LSB part to the precise MSB sub-adder.

The Generic Accuracy-Configurable Adder (GeAr) was proposed in [42]. It presents a fully configurable imprecise adder, where the number of sub-adder units can be selected and, for each sub-adder, the number of carry prediction bits, the number of sum bits and the bit width can be selected according to the application needs. This adder uses overlapped sub-adders.

The Lower-Part-OR Adder (LOA) was proposed in [43]. It splits the addition into two non-overlapped sub-adders. The MSB sub-adder does not use any imprecision technique and it is a conventional full adder. The imprecision is applied at the LSB sub-adder, which is significantly simplified. The carry propagation is eliminated in the LSB sub-adder and a simply bitwise OR is applied to the inputs. An extra AND is used in the most significant bits of this LSB adder to generate the carry-in for the MSB sub-adder, to reduce the imprecision [43].

3.2 Imprecise adders comparison and evaluation

A first evaluation of the considered imprecise adders was done to identify the configurations of these operators with higher potential to be applied in the SAD calculation of the HEVC intra-prediction.

This first evaluation considered 26 different configurations of these six imprecise adders. The evaluated adders were described in C++ and stimulated using 99,840 samples, extracted from the first frame from a class D test video sequence (*BasketballPass_416x240_50.yuv*). Since the video samples are 8-bit wide, the adders bit width was also defined as 8 bits. The imprecise adder results were also compared to a conventional (and precise) adder. The evaluation criteria were the following: average error and standard deviation. According to this preliminary evaluation, the configurations with the best results were: (1) ACA-II using the 4-bit overlapped sub-adders; (2) CCB using 2-bit sub-adders and one bit for the cut-back; (3) ETA-I using three precise and five imprecise bits; (4) ETA-IV using three sub-adders (3 bits, 3 bits, 2 bits), two bits in the first carry generation and three bits in the second carry generation; (5) GeAr using two 5-bit overlapped sub-adders; and (6) LOA using three precise and five imprecise bits. Table 1 shows a summary of this evaluation.

The second conducted evaluation identified the best configurations (among the six previously identified imprecise adders) that provide the best results in the particular context

Table 1 Hardware evaluation of the considered adder structures

Adder	Average error	Standard deviation
ACA-II	5.34	11.60
CCB	7.13	6.88
ETA-I	15.44	32.67
ETA-IV	1	0
GeAr	4.27	9.56
LOA	13.82	29.32

of the HEVC intra-prediction. For such purpose, the HEVC reference software (HM 16.2) [34] was used to measure the coding efficiency impacts. Hence, besides the original HM version, other six modified versions were generated, one for each previously presented imprecise adder. The imprecise operators were only considered in the first stage of the HM intra-prediction SAD operations to avoid accumulated error effects. This first SAD operation corresponds to the subtraction that is needed to generate the sum of absolute differences, as will be detailed in Sect. 5. The flag used in HM to enable (or not) the use of Hadamard in the intra-prediction module was disabled to guarantee that only SAD operations are enabled and SATD is not allowed [34].

The results of this experiment were evaluated using the output BD rate [27], which depicts the percentage increase (or decrease) in the number of bits that are necessary to represent the encoded video, considering the same objective image quality (PSNR). This experiment considered the Common Test Conditions (CTC) [33] defined by the HEVC community. Then, the 24 video sequences recommended by the CTC (with resolutions varying from 2560 × 1600 to 416 × 240 pixels) and four QP values (22, 27, 32 and 37) were used, giving rise to a total of 576 evaluations using the All Intra HM configuration [33].

The results of this experiment are presented in Table 2, considering the six classes of videos defined in the CTCs. According to these results, the lowest impacts in terms of average BD rate were obtained for ETA-IV and LOA adders, since these adders present the best results for all video classes.

A complimentary experiment was done to evaluate the power dissipation of these imprecise adders in a hardware implementation, as the focus of this work is to provide a low-energy and EQ-scalable solution. Note that an accurate energy estimation is not possible at this point, since it would require a full architectural (number of operators and parallelism) and data content (video resolution and frame rate) information. In accordance, these operators were evaluated by considering their average power dissipation. This power evaluation was done using more than two billion samples extracted from FourPeople test sequence [33].

Table 3 Hardware evaluation of the considered adder structures

Adder	Power (μW)	Area (Kgates)	Delay (ps)	PDP ($\times 10^{-3}$)
RCA	129	0.535	905	116.75
CLA	137	0.579	774	106.04
ACA-II	130	0.516	786	102.18
CCB	125	0.506	941	117.63
ETA-I	106	0.512	814	86.28
ETA-IV	134	0.549	829	111.09
GeAr	128	0.496	805	103.04
LOA	105	0.480	693	72.77

Table 3 presents the obtained results for the six considered imprecise adders and two precise adders—Ripple Carry Adder (RCA) and Carry-Lookahead Adder (CLA)—considering power dissipation, delay, silicon area, and Power–Delay Product (PDP).

Figure 2 represents these implementation results using radar charts, to facilitate this multi-variable comparison. The different axis reflects the percentage of increase or decrease in each criterion when compared to RCA and the smallest gray area depicts the best result when all compared variables are considered together. The BD-rate measures from the previous experiment were also inserted in these charts to allow a complete comparison.

According to these results, the best adder in all evaluated criteria was LOA, with outstanding results in delay and PDP. The GeAr adder posed in second place in terms of area usage, ACA-II reached the second in delay, ETA-I was the second in power consumption and in PDP. Some imprecise adders even reached worst results than RCA and CLA in some compared criteria, as presented in Table 3 (see PDP results). CLA presented the highest power dissipation and area but reduced delay. Therefore, its PDP was lower than some approximate adders (CCB and ETA-IV).

In accordance, one can conclude that LOA reached the best results when all compared variables are considered together, since it presented the smallest gray area among all the radar charts. In fact, although LOA has slightly worst BD rate than ETA-IV, the LOA hardware results are much better

Table 2 BD-rate increase as a result of using imprecise adders

	ACA-II (%)	CCB (%)	ETA-I (%)	ETA-IV (%)	GeAr (%)	LOA (%)
Class A	1.81	1.09	1.11	0.39	1.21	0.65
Class B	2.36	1.32	1.38	0.39	1.45	0.77
Class C	2.92	1.39	1.39	0.33	1.66	0.76
Class D	2.42	1.23	1.26	0.35	1.33	0.71
Class E	3.76	2.38	2.31	0.65	2.48	1.19
Class F	2.22	1.58	0.67	0.05	1.33	0.11
Average	**2.52**	**1.45**	**1.32**	**0.35**	**1.53**	**0.68**

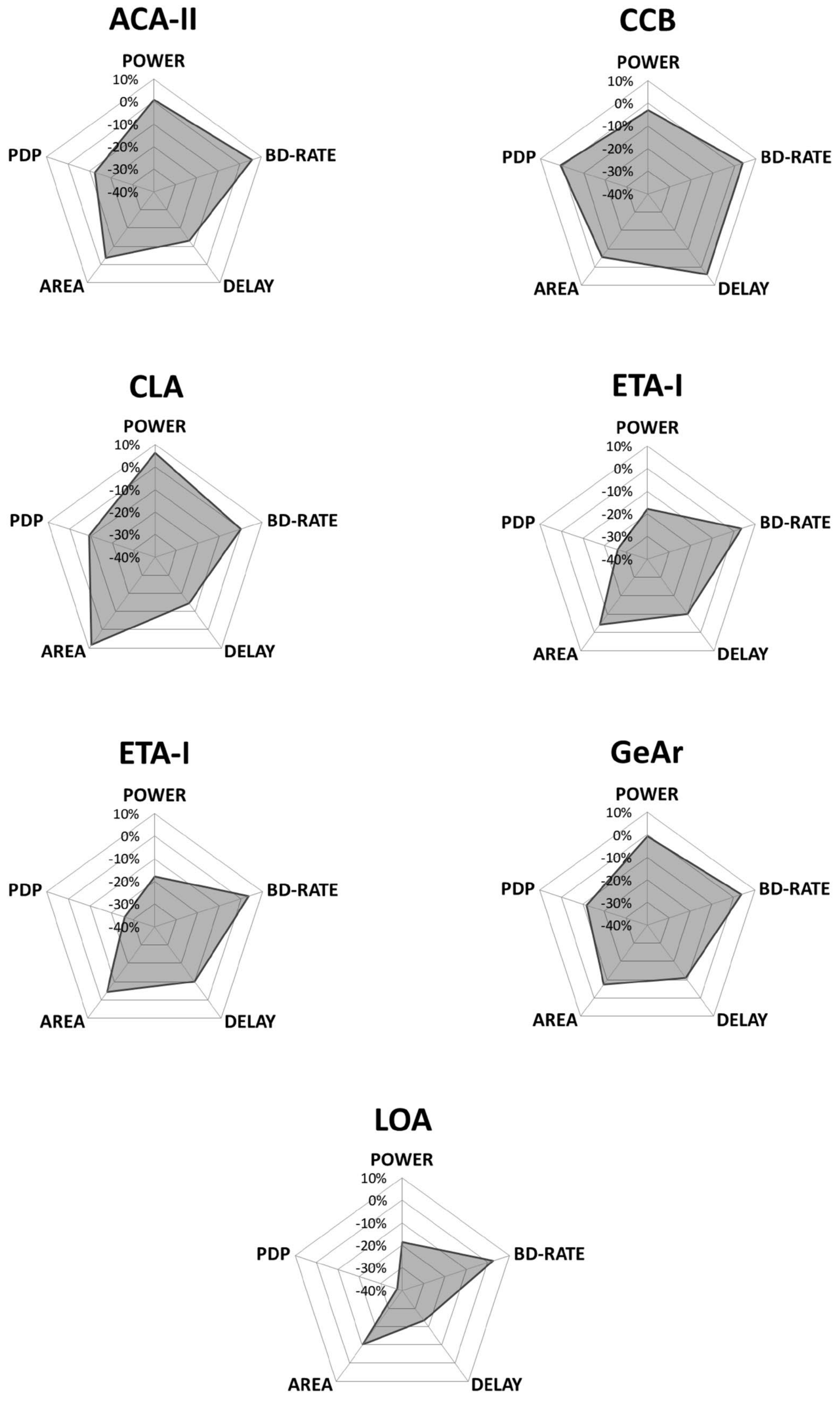


Fig. 2 Multi-variable comparison of the imprecise adders related to RCA

than ETA-IV for all considered criteria. LOA adders also have an interesting characteristic: the imprecision level can be changed. In fact, it is possible to design a configurable solution using multiple levels of imprecision, with distinct impacts in area, delay and, mainly, in power. This discussion will be detailed in the next section.

Hence, considering the described LOA features and the reached evaluation results, it was selected to be used in the architecture presented in this article. It presents a better support to design an energy-quality scalable SAD architecture and it reached the best results in terms of power consumption and delay, which is also important to process 8K videos in an energy-efficient way.

As mentioned before, the Lower-Part-OR Adder splits the addition into two non-overlapped sub-adders, as presented in Fig. 3. The MSB sub-adder does not use any imprecision technique and it is a conventional full adder. The imprecision is applied at the LSB sub-adder, which is significantly simplified. The carry propagation is eliminated in the LSB sub-adder and a simply bitwise OR is applied to the inputs. An extra AND is used in the most significant bits of this LSB adder to generate the carry-in for the MSB sub-adder, to reduce the imprecision [43].

4 Energy-quality scalable SAD unit

This section presents the proposed energy-quality scalable SAD Unit architecture. It was designed to be fully compliant with a previously proposed intra-prediction module [31], supporting all 35 intra-prediction modes and being able to process 64×64 CTBs. Each 64×64 CTB contains a total of 256, 64, 16, 4, and 1 blocks of sizes 4×4, 8×8, 16×16, 32×32, and 64×64, respectively. Hence, when considering a 64×64 CTB, a total of 341 individual blocks must be processed.

4.1 Base SAD unit architecture

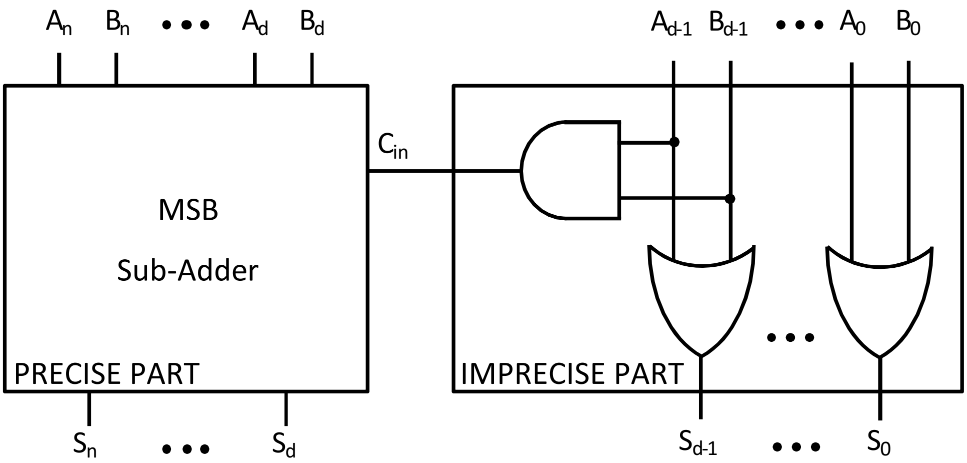
The block diagram of the SAD unit that was proposed in [31] is presented in Fig. 4. This architecture was designed to process ultra-high-resolution videos in real time, by supporting the encoding of UHD 8K videos. To allow this very high throughput, the architecture makes use of 35 parallel SAD trees. Each SAD tree can process one 16×16, 8×8 or 4×4 input block in only one single clock cycle; the 32×32 blocks are processed in four cycles and the 64×64 blocks are processed in 16 cycles.

The external interface of this SAD unit simultaneously receives the 8-bit input samples from the original block and the corresponding samples of the 35 predicted blocks. Each SAD tree receives a different candidate block, but all SAD trees receive the same block to be predicted. The block size control is done through the SelBlock signal. This unit also outputs the SADs of the 35 predicted blocks, using 16 bits.

Figure 4 details its internal architecture, composed of an array of SAD trees, numbered from 0 to 34, corresponding to each intra-prediction mode. The SelBlock signal controls a MUX responsible to select blocks of size 4 × 4, 8 × 8, 16 × 16, and ¼ of a 32 × 32 when the signal values are "00", "01", "10", and "11", respectively.

Considering the very high level of parallelism that is adopted in this intra-prediction architecture, the processing of a complete CTB composed of the 341 candidate blocks requires a total of 368 clock cycles. Since each UHD 8K video frame (7680×4320 pixels) includes a total of 12,150 64×64 CTBs (considering a 4:2:0 color subsampling [32]), one frame is processed in 4,471,200 clock cycles and a minimum operation frequency of 268.3 MHz is required to process 60 frames per second.

Fig. 3 Lower-Part-OR Adder structure



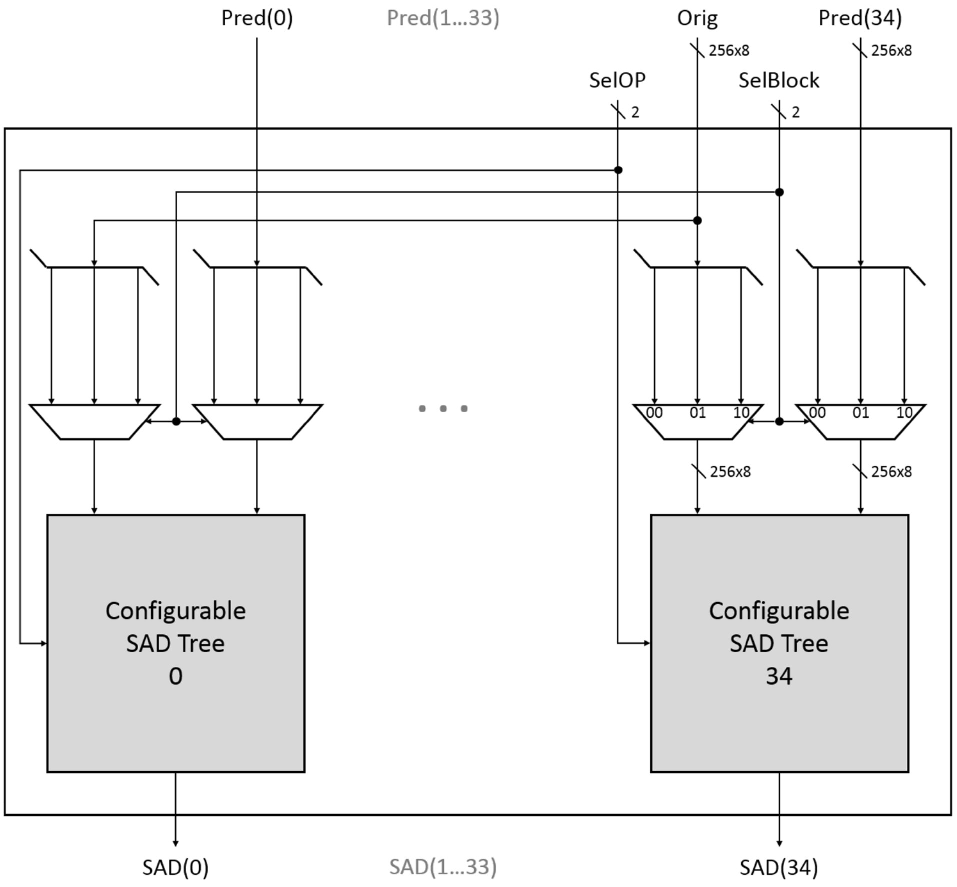


Fig. 4 Block diagram of the configurable SAD unit

4.2 Imprecise operation points definition

The preliminary evaluations that were conducted in Sect. 3 identified LOA as the best option to implement the aimed energy-quality scalable intra-prediction SAD unit architecture. In this section, a new set of experiments is considered to define the imprecise operation points of this architecture. For such purpose, eight SAD tree configurations were considered, corresponding to multiple levels of imprecision and energy-quality scalability. Once again, and similarly to the discussion presented in Sect. 3, a power characterization of the basic SAD units is used to make decisions towards a low-energy and EQ-scalable architecture. In this evaluation, the considered SAD tree configurations can process, in parallel, a complete 16 × 16 samples block. Hence, each SAD tree input is formed by 256 samples of the current block and 256 samples of the predicted block.

Since the SAD tree architecture has 8-bit inputs, eight operating points were considered: without imprecision and with 1 bit to 7 bits of imprecision. As discussed before, the imprecision was inserted only in the first stage of the SAD calculations, i.e., in the subtraction and modulo operations, using LOA structures. The accumulation of the remaining SAD tree layers is performed by conventional RCA operators to avoid error accumulation. Each SAD tree has nine levels of combinational operations and the number of arithmetic operators per level is 256, 128, 64, 32, 16, 8, 4, 2 and 1. The SAD tree architectures were designed to evaluate the power and area gains of each imprecision level when compared with the precise version. These architectures were described in VHDL and the synthesis considered the same methodology that was presented in Sect. 3.

These seven imprecise SAD calculation setups were also implemented in a modified HM reference software to evaluate the coding efficiency impacts of each imprecision

Table 4 SAD trees syntheses and BD-rate results

	Power (mW)	Area (Kgates)	BD-rate increase (%)	Power reduction	Area reduction
RCA	**17.70**	**31.68**	**0**	–	–
LOA1	16.73	31.33	0.27	5.48%	1.10%
LOA2	16.73	31.34	0.26	5.48%	1.07%
LOA3	**15.71**	**29.65**	**0.28**	**11.24%**	**6.41%**
LOA4	14.94	28.65	0.40	15.59%	9.56%
LOA5	**13.83**	**26.71**	**0.68**	**21.86%**	**15.69%**
LOA6	13.10	26.25	1.14	25.99%	17.14%
LOA7	**11.93**	**24.47**	**1.72**	**32.60%**	**22.76%**

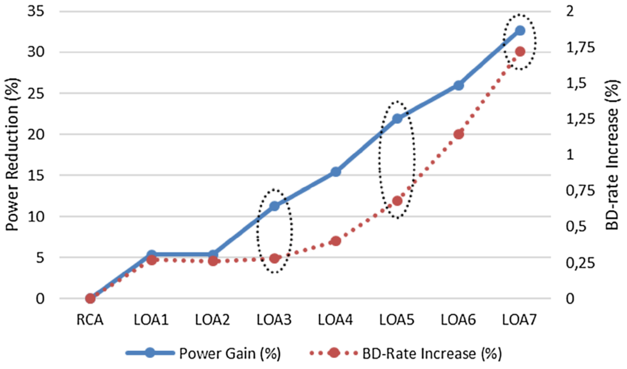


Fig. 5 Power reduction (%) vs. BD-rate increase (%)

level. This evaluation used the same methodology shown in Sect. 3. The obtained synthesis and HM evaluation results are presented in Table 4. RCA refers to the precise version. Imprecise versions are referred to as LOA and the number following the abbreviation indicates the number of imprecise bits that are used in the arithmetic operators. In general, the higher is the imprecision level, the lower is the area and power dissipation, and the higher is the BD-rate degradation.

The obtained results, corresponding to the relation between power reduction and BD-rate increase, were used to define the set of operation points to be considered in the SAD unit. Figure 5 shows a chart that presents this relation. The final decision was to include LOA3, LOA5, and LOA7 as the imprecise operation points of the SAD tree architecture, since power dissipation reductions are meaningful and close to 10%, 20%, and 30%. The BD-rate increase for these three imprecise operation points was 0.3%, 0.7%, and 1.7%, respectively.

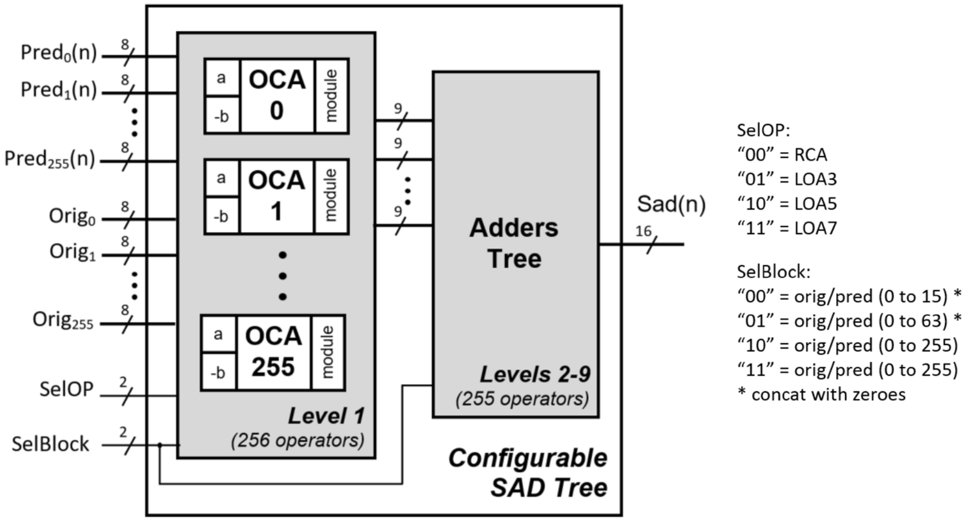
These operation points are highlighted in Table 4 and in Fig. 5. The reached results showed that impressive power reductions can be obtained with very low coding efficiency losses by considering the application target and the device status. A fourth architectural operation point was defined as the precise version, without any coding efficiency losses.

4.3 Energy-quality scalable SAD tree architecture

The proposed SAD tree architecture uses the same architectural template used in the base intra-prediction architecture [31]. It was designed to process 512 input samples in parallel, 256 from the original block and 256 from the predicted block. The architecture is fully combinatorial. This means that the SAD of a 16 × 16 predicted block is calculated in one single clock cycle. The main difference between the base and the newly proposed architecture is in the first level of the SAD units: while the base SAD tree architecture uses RCAs to calculate the module of differences, the proposed SAD tree uses a new configurable operator, denoted as Optimized Configurable Adder (OCA), that will be presented in the next paragraphs.

A straightforward and non-optimized approach to design this configurable SAD architecture would simply instantiate the four selected imprecise adder tree architectures (RCA, LOA3, LOA5, and LOA7). The output of these four SAD trees could be connected through a multiplexer, with the output depending on the selected operation mode. However, besides using a large amount of hardware resources, this non-optimized solution would also tend to increase the power dissipation, since all adders of all SAD trees would switch at each new input. Hence, since the complete intra-prediction architecture uses 35 SAD calculation trees, the application of further optimizations is even more important to reduce the power dissipation and area. Thus, an optimized SAD calculation tree was redesigned, by exploiting the sharing of common operations and operand isolation techniques.

Fig. 6 Block diagram of the configurable SAD tree architecture



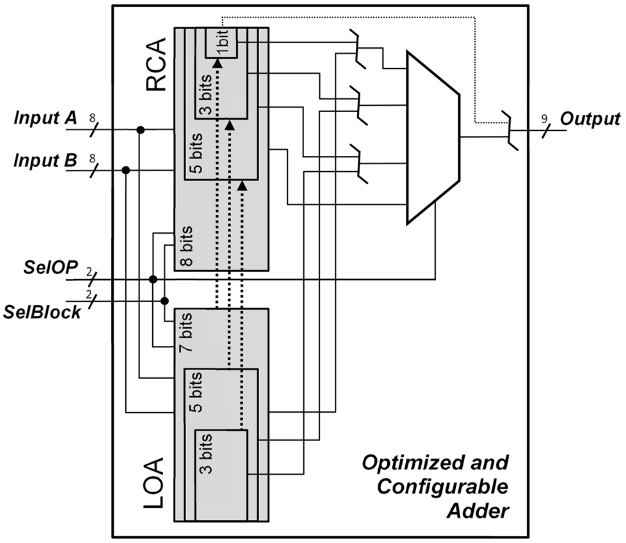


Fig. 7 Block diagram of the optimized and configurable adder

Figure 6 shows a high-level block diagram of the configurable and optimized SAD calculation tree that is now proposed. The most important element in this architecture is the Optimized Configurable Adder (OCA) structure that will be detailed in the next paragraphs. The Orig and Pred(n) inputs refer to the 8-bit samples from the original block and from one of the n candidate blocks. The output Sad(n) is the computed SAD for the predicted block n, using 16 bits. The SelBlock input has the same purpose as defined in the base SAD Unit architecture and the SelOp input selects the desired operation point.

The first level of the SAD tree is implemented with 256 OCA units, which implement a subtraction followed by the module calculation, considering the four operation points defined in the previous subsection. Both operations were grouped to reduce the hardware consumption, using a combinatorial logic based on a Carry-Lookahead Adder [44] with several simplifications, especially exploring the LOA behavior. The Adders Tree block in Fig. 6 is responsible to levels 2–9 of SAD calculations and it is used to accumulate the absolute differences. As discussed in Sect. 3, this adders tree employs RCA adders to avoid accumulated error effects.

The main idea that is explored in the OCA operator is the reuse of as many bits as possible of the RCA and LOA structures. Figure 7 presents its block diagram. The dotted lines represent the carry propagation. This solution uses only an 8-bit RCA, a 7-bit LOA, and some additional logic to control the conditional carry propagation that supports the four operation points defined in this article. Other additional logic is required to organize the outputs, concatenating the adequate LOA output bits with the adequate RCA output bits to reach each configurable imprecision level. LOA3, LOA5, and LOA7 operation points share the three LOA less significant bits and LOA5 and LOA7 also share other two LOA bits, as presented in Fig. 7. The same behavior occurs with RCA. As an example, LOA5 operation point will use five bits from the LOA operator and three bits from the RCA operator. The partial results are appropriately concatenated to generate the operator results.

Hence, while the non-optimized solution would require four independent 8-bit adders for each of the 256 operations, corresponding to one RCA, one LOA3 (5-bit RCA plus 3-bit LOA), one LOA5 (3-bit RCA plus 5-bit LOA) and one LOA7 (1-bit RCA plus 7-bit LOA); the proposed optimized unit saves nine bits of RCA and eight bits of LOA, corresponding to 52.9% of the RCA bits and 53.3% of the LOA bits. These savings in area (and power) are especially important when considering that the SAD tree uses 256 of such operators.

The use of the OCA structure also allows an easy sharing of the other SAD tree levels (2–9 in Fig. 6). In other words, the same Adder Tree structure is used to all operation points. Hence, only 255 operators are used in levels 2–9 of the optimized SAD tree architecture, saving 765 adders. Considering the number of adder bits, this solution uses 2542 bits of addition, instead of 10,168. This means that 75% of the operators were saved in the Adders Tree.

Hence, when considering the whole SAD structure, the proposed optimized unit requires 4590 RCA bits and 1792 LOA bits, instead of 14,520 RCA bits and 3840 LOA bits that would be required without optimizations. This means that the proposed optimized SAD tree saves 68.4% of RCA bits and 53.3% of LOA bits. These expressive savings in hardware resources are especially important when considering that 35 SAD trees are used in the SAD unit. Naturally, these area savings result in similar impacts in power dissipation.

The operand isolation [45] technique was also applied to further optimize the power dissipation, by isolating the operators that are not used at each calculation. This isolation is controlled by signals generated from SelBlock and SelOp, presented in Figs. 6 and 7, and it is applied through an extra AND gate inserted at each adder input.

The application of operand isolation considered two situations. The first is when the operation point is selected and affects only the first SAD tree level, which is the configurable part of this architecture. In this case, since the OCA operators are optimized at bit level, the operand isolation is also applied at bit level and the 1-bit operators that are not necessary for each operation point are isolated. The second situation occurs over all nine levels of SAD tree architecture whenever smaller block sizes are processed (8 × 8 or 4 × 4) and a part of the adders is not necessary. As an example,

Table 5 Optimized and configurable SAD tree architecture results

Evaluation criteria	Original SAD tree	Optimized and configurable SAD tree			
		RCA	LOA3	LOA5	LOA7
Area (Kgates)	38.1	43.9			
Area increase		15.0%			
Power (mW)	20.3	14.9	14.1	13.4	12.8
Power reduction	–	26.5%	30.3%	33.8%	36.8%

Table 6 Energy-quality scalable SAD unit results

Evaluation criteria	Original SAD unit	Energy-quality scalable SAD unit			
		RCA	LOA3	LOA5	LOA7
Area (Kgates)	1288.7	1388.3			
Area increase	–	7.7%			
Power (mW)	692.3	505.3	481.2	461.3	443.1
Energy/frame (mJ)	11.53	8.42	8.02	7.68	7.38
Energy reduction	–	27.0%	30.5%	33.4%	36.0%
BD-rate losses	0%	0%	0.28%	0.68%	1.72%

when an 8 × 8 block is processed, only 127 outputs of the SAD tree operators are necessary.

The additional hardware (and the consequent power overhead) that is introduced by the operand isolation technique is widely justified, since the SelBlock and SelOp control signals tend to be stable for a high number of input blocks during the video encoding process.

5 Experimental evaluation

The proposed EQ-scalable SAD unit intra-prediction architecture was fully described in VHDL and synthesized using the Cadence Encounter RTL compiler tool, targeting the Nangate standard cells library for 45 nm technology at 1.1 V [46]. To guarantee real-time processing of 8K UHD resolution (7680 × 4320 pixels) at 60 fps, the target operating frequency was defined as 269 MHz. The required hardware resources are presented in equivalent nand2 gates, obtained by dividing the total circuit area by the area of a nand2 cell (0.8 μm^2) in this technology. The power dissipation results considered a switch activity of 25%.

5.1 Optimized and configurable SAD tree

Table 5 presents the area and power consumption results of the configurable SAD tree architecture, when compared with the original non-configurable base structure (using only RCAs). The power dissipation results are presented for each of the defined operating points for the optimized architecture.

The presented results emphasize the significant reductions in power dissipation that are obtained with the optimized architecture when compared to the original base version. These power reductions, varying from 26.5 to 36.8%, arise from the adoption of operand isolation techniques. Despites supporting four operation points and using the extra hardware required to implement the operand isolation, the configurable and optimized architecture used only 15% more area than the original version. Even with this extra area, the power gains were expressive.

5.2 Energy-quality scalable SAD unit

This subsection discusses the obtained experimental results after the implementation of the proposed energy-quality scalable SAD unit. This structure simultaneously processes all prediction modes (and block sizes) defined by the HEVC intra-prediction specification.

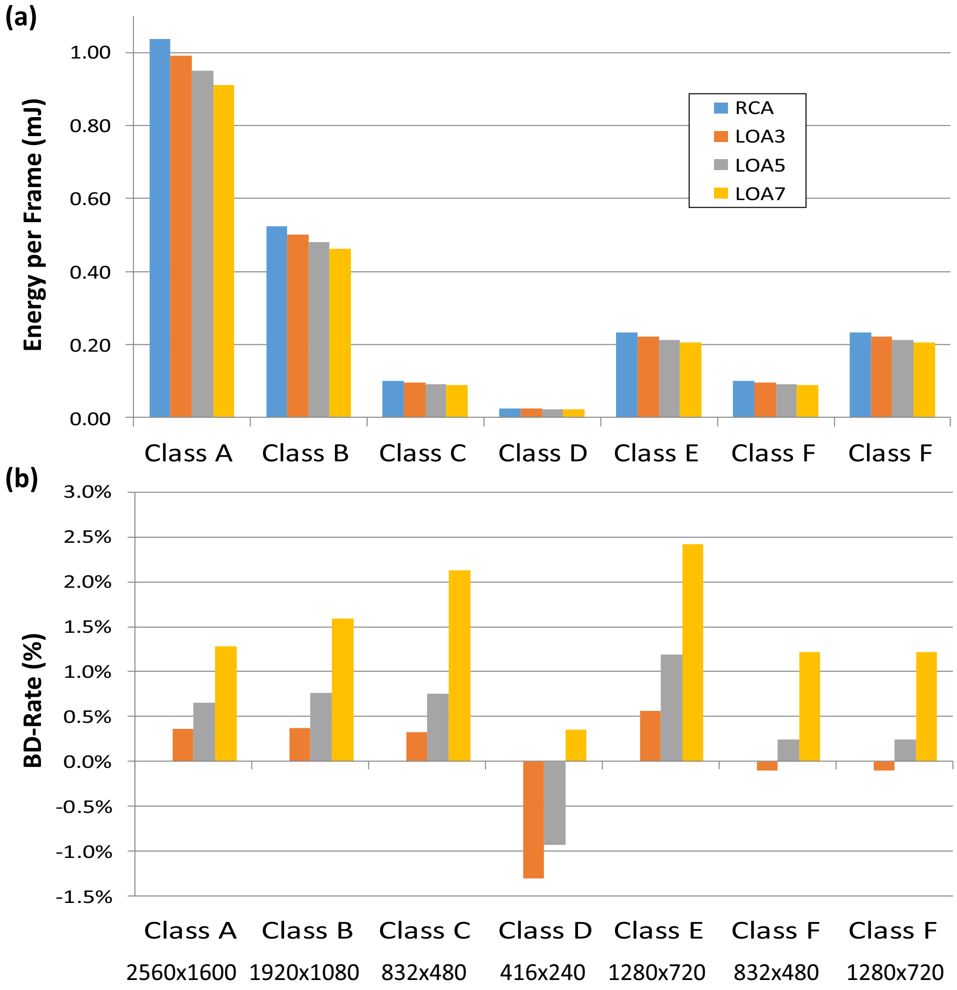
Table 6 presents the gate count, power dissipation (considering a switch activity of 25%), consumed energy (to process one 8K UHD frame), and coding efficiency degradation in terms of BD rate (used to measure the application quality, as discussed in Sect. 1) for the original (precise) architecture and for the proposed EQ-scalable architecture, when running on each operation point: RCA (precise mode), LOA3, LOA5, and LOA7.

The obtained results show that the proposed SAD Unit architecture provides a reduction of the consumed energy between 27 and 36%, when compared to the original architecture. One can also observe that even the precise solution (RCA) is more power efficient than the original architecture, as a result of the application of the operand isolation technique. Hence, the total energy required to process a UHD 8K frame is reduced from 11.53 mJ down to 8.42 mJ, considering a precise computation. When employing the considered approximate operation points, the energy consumption drops to 8.02 mJ (LOA3), 7.68 mJ (LOA5) and 7.38 mJ (LOA7). Note that power and energy are proportional in this case, as the throughput/frame processing time remains constant for the same resolution and frame rate. These energy savings come at the cost of a consequent BD-rate increase, ranging from 0.28 (LOA3) to 1.72% (LOA7), and of an area increase of 6.8%. These BD-rate results were obtained according to the CTC [33].

5.3 Energy-quality scalability

A detailed characterization relating the energy consumption reduction and the consequent BD-rate variation for different video resolutions and imprecise operation points is presented in Fig. 8. The average results of each class were

Fig. 8 Energy (**a**) and BD-rate (**b**) variations for different resolutions and different operation points



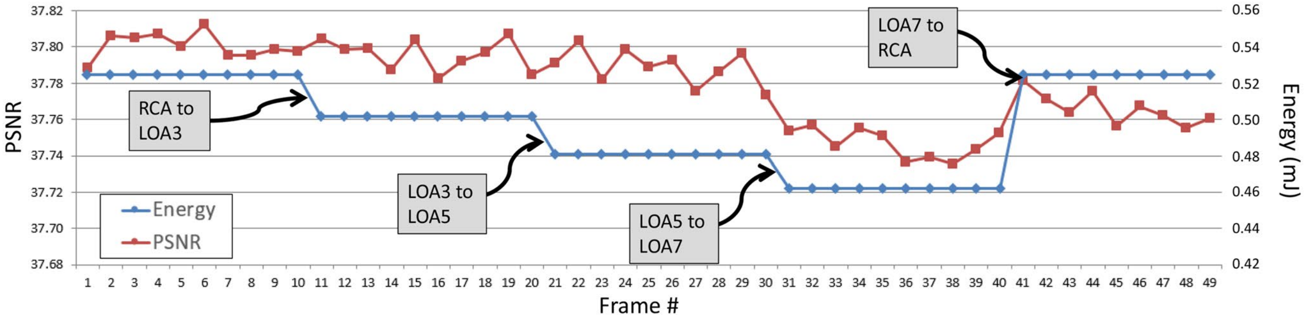


Fig. 9 Energy and video quality variation along the time for different operation points

considered for all the videos recommended in the CTC. In these experiments, the proposed architecture executes at its maximum operating frequency (269 MHz). From Fig. 8a, one can observe that a Class A video frame (2560 × 1600) requires about 1.03 mJ to be processed using RCA adders. This consumed energy can be reduced to 0.91 mJ (− 12%) by adopting imprecise calculations with the LOA7 operation point, at the cost of a slight increase (1.28%) of BD rate. LOA3 and LOA5 operation points represent, respectively, an energy reduction of 4% and 9% when compared to the RCA-based prediction.

As expected, lower resolutions demand less energy per frame due to the lower amount of data to be processed. However, no clear relation between video resolution and coding efficiency was observed, as it can be observed in Fig. 8b. This behavior is probably explained because the impact of imprecision depends on the video content (texture) rather than on video resolution. In turn, the relation between the imprecise operation points is consistent across all video resolutions. Furthermore, besides providing a very low BD-rate increase for most setups, LOA3 even presents some coding efficiency gains for Classes D (416 × 240) and F (832 × 480 and 1280 × 720): about 1.3% BD-rate reduction for Class D. On the opposite side, LOA7 introduces the greatest losses, ranging from 0.35 (Class D) up to 2.41% (Class E).

Whereas Fig. 8 demonstrates the scalability range in terms of the introduced imprecision and video resolution, Fig. 9 shows the EQ behavior along the time when the operation point is dynamically modified. In this experiment, 50 frames from the BQTerrace video sequence were encoded at QP 27, according to the following conditions: frame 0–9 using the RCA configuration, frames 10–19 using the LOA3 configuration, fames 20–29 using LOA5, frames 30–39 using LOA7 and, finally, frames 40–49 using the RCA configuration. The energy consumption plateaus, defined by each operation point, are easily observed. In turn, the video quality presents a greater variation due to changes of video content. Still, it is possible to observe that the resulting video quality degradation is very small (PSNR reducing from 37.8 to 37.74 dB) observed when the imprecision level was at its maximum (especially for LOA 7; frames 30–39). Since there are no dependencies between frames, as soon as the operation mode changes from LOA7 to RCA, a video quality increase is immediately observed (see frame 40 in Fig. 9). This behavior demonstrates that the proposed EQ-scalable hardware architecture can be easily adjusted by an external controller (e.g., battery level monitor) and it is suitable for integration in a complete video encoder system.

6 Related works

The literature presents a few published works with hardware results for the HEVC intra-prediction encoder, but none of them refers to an energy-quality scalable architecture. There are also some other works targeting the HEVC decoder, such as [47–51], but the decoder does not require SAD calculations, making any comparison with the proposed unit impractical. Actually, only a few published works targeting the encoder present power or energy results and, in general, the diversity of design options prevents a fair comparison.

Works like [52, 53] use SATD in their architectures (instead of SAD), making a fair comparison with the presented work unfeasible. Despite focusing at the intra-prediction encoder, works like [54–56] only present the details about the designed hardware for the block prediction and do not discuss the introduced distortion.

Despite targeting a different standard-cells technology, the works [8–10] use SAD as distortion criterion and some comparisons are actually possible. But it is important to emphasize that none of these works present power or energy results, nor they present independent results for the SAD unit.

The hardware presented in [8] has a parallelism of 16 samples per cycle, uses pipeline and can process Full HD videos at 30 fps. The necessary operation frequency to reach this processing rate is 600 MHz, which is more than twice higher than the operation frequency that is necessary by the proposed structure to process UHD 8K videos. On the other hand, such work used only 77Kgates, which is much less than the area required by the developed architecture. The applied simplifications cause a little drop of 0.13% in BD rate.

The architecture proposed in [9] has a parallelism level of 64 samples per cycle and uses an alternative processing order to reduce the memory accesses. This hardware can process Full HD videos at 60 fps, running at 400 MHz and using 324Kgates. Despite using less hardware than the presented structure, the processing rate reached by [9] is much lower, requiring a higher frequency to support a lower resolution.

The work in [10] not only focuses on an efficient memory hierarchy targeting the intra-prediction, but also it presents a dedicated hardware structure. This hardware has a parallelism level of eight samples per cycle and runs at 500 MHz, reaching a throughput able to process HD 720 p videos at 30 fps. This work also requires a higher operation frequency than the proposed architecture to process lower resolutions, as a result of the lower parallelism that is exploited. This lower parallelism also allows the hardware presented in this work to use only 36.7Kgates.

In [57], a high-throughput SAD implementation targeting motion estimation is presented. The authors use carry-save adder (CSA) trees to compress the absolute differences. Synthesis results for a 180-nm technology report a 12.5% delay improvement and 9% area reduction when compared to a baseline CLA-based architecture. Although it is fair to assume that [57] will lead to some power reduction, no power analysis is provided. In turn, the presented solution reaches up to 36.8% of power reduction. Moreover, since the imprecise modules are restricted to the first level of adders (where the differences are calculated—see Fig. 6) and the solution in [57] focuses on the accumulation levels located after absolute operators (levels 2–9 in the proposed architecture—see Fig. 6), both works can be deployed together to deliver further improvements.

The work described in [58] proposes a low-cost SAD architecture adopting 4–2 compressors. Compared to RCA-based architectures, 42–48% delay reduction is observed at the cost of 31–39% area increase. Conversely, when compared with a baseline RCA structure, the proposed LOA operators reduce the delay with a 7.7% area increase. The achieved power reduction ranges from 27 to 36%, whereas [58] does not report power or energy results.

In [59], the authors propose a SAD operator that compresses the propagated data and optimizes the adder trees. Synthesis results for TSMC 180 nm show that a 12.1% area reduction is obtained when compared to a straightforward SAD implementation. The solution in [59] dissipates 461.5 mW at 227 MHz to process VGA motion estimation in the context of H.264. The proposed intra-prediction unit dissipates 443–505 mW to process the intra-prediction for UHD 8K. These numbers are indicators of the efficiency of the presented solution. However, it is not possible to directly compare the related works since none present an energy-quality scalable solution and they differ in terms of the target application (the SAD operators are used in different encoder units), video encoding standard and ASIC technology.

7 Conclusions

This article presented an energy-quality scalable SAD unit architecture targeting the HEVC intra-prediction of UHD 8K videos. The energy-quality scalability was reached using approximated computing implemented with imprecise adders in four distinct operation points.

Six imprecise adders were evaluated, considering its coding efficiency and hardware results, leading to the selection of LOA as the most convenient structure to be used in the designed architecture. The definition of the operation points was done by evaluating several independent implementations of SAD tree architectures. Through this evaluation, four operation points were defined: RCA, LOA3, LOA5 and LOA7.

The designed SAD unit used a configurable SAD tree based on an optimized and configurable adder structure that supports the four operation points. This optimized adder (and the complete SAD tree) also used operand isolation technique, to reduce the power dissipation. The proposed SAD unit uses 35 parallel instances of the configurable SAD to reach the desired throughput.

The energy-quality scalable SAD unit is able to implement the HEVC intra-prediction of UHD 8K videos at 60 frames per second running at 269 MHz and considering four operation points. When compared with the previous architecture [31], it reduces the required energy to process one UHD 8K frame from 11.53 mJ down to 8.42 mJ (about 27%). Such energy reduction comes at the cost of a slight coding efficiency loss of 0.28%, 0.68%, and 1.72% for the three approximate operation points. Additional experiments demonstrated that the proposed architecture allows EQ scalability for different resolutions and frame rates and can be controlled by an external controller, being suitable for integration in an EQ-scalable video encoder system.

Finally, it should be noted that the applied methodology and the set of optimizations proposed to the SAD unit are applicable in other encoding steps of the video encoder (such as motion estimation), increasing its potential EQ scalability. Furthermore, it can also be used in other image/video processing and computer vision algorithms that use SAD as similarity criterion. Moreover, the OCA operators can be employed to optimize the computation of other widely used criteria such as SATD and SSE.

Acknowledgements 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/50021/2019, and also by CNPq and FAPERGS Brazilian research support agencies.

References

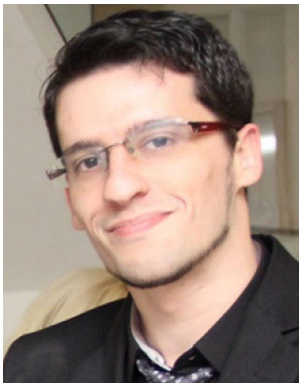
1. Cisco Visual Networking Index: Forecast and Trends, 2017–2022. Cisco Systems. San Jose, USA [Online]. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html. Accessed 23 Apr 2019
2. Information Technology.: High efficiency coding and media delivery in heterogeneous environments—part 2: high efficiency video coding, ISO/IEC 23008-2 (2013)
3. Series H.: Audiovisual and multimedia systems infrastructure of audio-visual services–advanced coding of moving video advanced video coding for generic audiovisual services, recommendation ITU-T H.264 (06/2011), (2011)
4. Correa, G., Assuncao, P., Agostini, L., Cruz, L.: Performance and computational complexity assessment of high-efficiency video encoders. IEEE Trans. Circuits Syst. Video Technol. **22**(12), 1899–1909 (2012). https://doi.org/10.1109/TCSVT.2012.2223411
5. Alcocer, E., Gutierez, R., Lopez-Granado, O., Malumbres, M.: Design and implementation of an efficient hardware integer motion estimator for an HEVC video encoder. J. Real Time Image Proc. **16**(2), 547–557 (2019). https://doi.org/10.1007/s11554-016-0572-4
6. Lung, C.-Y., Shen, C.-A.: Design and implementation of a highly efficient fractional motion estimation for the HEVC encoder. J. Real Time Image Process. **16**, 1–17 (2016). https://doi.org/10.1007/s11554-016-0663-2
7. Paim, G., Penny, W., Goebel, J., Afonso, V., Susin, A., Porto, M., Zatt, B., Agostini, L.: An efficient sub-sample interpolator hardware for VP9-10 standards. In: IEEE International Conference on Image Processing, pp. 2167–2171. Phoenix, USA (2016). https://doi.org/10.1109/icip.2016.7532742
8. Liu, C., Shen, W., Ma, T., Fan, Y., Zeng, X.: A highly pipelined VLSI architecture for all modes and block sizes intra prediction in HEVC encoder. In: IEEE International Conference on ASIC, pp. 1–4. Shenzhen, China (2013). https://doi.org/10.1109/asicon.2013.6811849
9. Zhou, N., Ding, D., Yu, L.: On hardware architecture and processing order of HEVC intra prediction module. In: Picture Coding Symposium, pp. 101–104. San Jose, USA (2013). https://doi.org/10.1109/pcs.2013.6737693
10. Palomino, D., Sampaio, F., Agostini, L., Bampi, S., Susin, A.: A memory aware and multiplierless VLSI architecture for the complete intra prediction of the HEVC emerging standard. In:

IEEE International Conference on Image Processing, pp. 201–204. Lake Buena Vista, USA (2012). https://doi.org/10.1109/icip.2012.6466830

11. Jridi, M., Alfalou, A., Meher, P.: Efficient approximate core transform and its reconfigurable architectures for HEVC. J. Real Time Image Process. (2018). https://doi.org/10.1007/s11554-018-0768-x
12. Braatz, L., Agostini, L., Zatt, B., Porto, M.: A multiplierless parallel HEVC quantization hardware for real-time UHD 8K video coding. In: IEEE International Conference on Circuits and Systems, pp. 1–4. Baltimore, USA (2017). https://doi.org/10.1109/iscas.2017.8050704
13. Goebel, J., Paim, G., Agostini, L., Zatt, B., Porto, M.: An HEVC multi-size DCT hardware with constant throughput and supporting heterogeneous CUs. In: IEEE International Conference on Circuits and Systems, pp. 2202–2205. Montreal, Canada (2016). https://doi.org/10.1109/iscas.2016.7539019
14. Jo, H., Park, S., Sim, D.: Parallelized deblocking filtering of HEVC decoders based on complexity estimation. J. Real Time Image Proc. **12**(2), 369–382 (2016). https://doi.org/10.1007/s11554-015-0556-9
15. Shen, W., Fan, Y., Bai, Y., Huang, L., Shang, Q., Liu, C., Zeng, X.: A combined deblocking filter and SAO hardware architecture for HEVC. IEEE Trans. Multimed. **18**(6), 1022–1033 (2016). https://doi.org/10.1109/TMM.2016.2532606
16. Rediess, F., Agostini, L., Cristani, C., Dall'Oglio, P., Porto, M.: High throughput hardware design for the adaptive loop filter of the emerging HEVC video coding. In: Symposium on Integrated Circuits and Systems Design, pp. 1–5. Brasília, Brazil (2012). https://doi.org/10.1109/sbcci.2012.6344446
17. Choi, J.-A., Ho, Y.-S.: High throughput entropy coding in the HEVC standard. J. Signal Process. Syst. **81**(1), 59–69 (2015). https://doi.org/10.1007/s11265-014-0900-5
18. Sun, H., Zhou, L., Xu, H., Sun, T., Wang, Y.: A high-efficiency HEVC entropy decoding hardware architecture. In: International Conference on Advanced Communication Technology (ICACT), pp. 186–190. Seoul, South Korea (2015). https://doi.org/10.1109/icact.2015.7224781
19. Ramos, F., Goebel, J., Zatt, B., Porto, M., Bampi, S.: Low-power hardware design for the HEVC binary arithmetic encoder targeting 8K videos. In: Symposium on Integrated Circuits and Systems Design, pp. 1–6. Belo Horizonte, Brazil (2016). https://doi.org/10.1109/sbcci.2016.7724044
20. Afonso, V., Maich, H., Agostini, L., Franco, D.: Low cost and high throughput FME interpolation for the HEVC emerging video coding standard. In: Latin American Symposium on Circuits and Systems, pp. 1–4. Cusco, Peru (2013). https://doi.org/10.1109/lascas.2013.6519017
21. He, G., et al.: High-throughput power-efficient VLSI architecture of fractional motion estimation for ultra-HD HEVC video encoding. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **23**(12), 3138–3142 (2015). https://doi.org/10.1109/tvlsi.2014.2386897
22. He, Z., Tsui, C., Chan, K., Liou, M.: Low-power VLSI design for motion estimation using adaptive pixel truncation. IEEE Trans. Circuits Syst. Video Technol. **10**(5), 669–678 (2000). https://doi.org/10.1109/76.856445
23. Yang, Y., Zheng, J.: Edge-guided depth map resampling for HEVC 3D video coding. In: International Conference on Virtual Reality and Visualization, pp. 132–137. Xi'an, China (2013). https://doi.org/10.1109/icvrv.2013.29
24. Masera, M., Martina, M., Masera, G.: Adaptive approximated DCT architectures for HEVC. IEEE Trans. Circuits Syst. Video Technol. **27**(12), 2714–2725 (2017). https://doi.org/10.1109/tcsvt.2016.2595320
25. El-Harouni, W., et al.: Embracing approximate computing for energy-efficient motion estimation in high efficiency video coding. In: Design, Automation and Test in Europe Conference and Exhibition (DATE), pp. 1384–1389. Lausanne, Switzerland (2017). https://doi.org/10.23919/date.2017.7927209
26. Porto, R., Agostini, L., Zatt, B., Porto, M., Roma, N., Sousa, L.: Energy-efficient motion estimation with approximate arithmetic. In: International Workshop on Multimedia Signal Processing, pp. 1–6. Luton, UK (2017). https://doi.org/10.1109/mmsp.2017.8122248
27. Bjontegaard, G.: Calculation of average PSNR differences between RD-curves. In: Document VCEG-M33. ITU—Telecommunications Standardization Sector—STUDY GROUP 16 Question 6—Video Coding Experts Group (VCEG) (2001). http://wftp3.itu.int/av-arch/video-site/0104_Aus/VCEG-M33.doc. Accessed 29 Mar 2019
28. Raha, A., Jayakumar, H., Raghunathan, V.: A power efficient video encoder using reconfigurable approximate arithmetic units. In: International Conference on VLSI Design and 2014 13th International Conference on Embedded Systems, pp. 324–329. Mumbai, India (2014). https://doi.org/10.1109/vlsid.2014.62
29. Jridi, M., Meher, P.: Scalable approximate DCT architectures for efficient HEVC-compliant video coding. IEEE Trans. Circuits Syst. Video Technol. **27**(8), 1815–1825 (2017). https://doi.org/10.1109/tcsvt.2016.2556578
30. Lainema, J., Bossen, F., Han, W., Min, J., Ugur, K.: Intra coding of the HEVC standard. IEEE Trans. Circuits Syst. Video Technol. **22**(12), 1792–1801 (2012). https://doi.org/10.1109/tcsvt.2012.2221525
31. Corrêa, M., Zatt, B., Porto, M., Agostini, L.: High-throughput HEVC intrapicture prediction hardware design targeting UHD 8K videos. In: IEEE International Symposium on Circuits and Systems, pp. 1–4. Baltimore, USA (2017). https://doi.org/10.1109/iscas.2017.8050702
32. Wien, M.: High Efficiency Video Coding: Coding Tools and Specification, pp. 63–65. Springer, New York (2014)
33. Bossen, F.: Common test conditions and software reference configurations. In: "Document JCTVC-L1100 of JCT-VC", Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Jan. 23 (2013). http://phenix.it-sudparis.eu/jct/doc_end_user/current_document.php?id=7281. Accessed 29 Mar 2019
34. "HEVC Reference Software". Fraunhofer Heinrich Hertz Institute. Berlin, Germany [Online]. https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/ Accessed 23 Apr 2019
35. Sullivan, G., Ohm, J., Han, W., Wiegand, T.: Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. Circuits Syst. Video Technol. **22**(12), 1649–1668 (2012). https://doi.org/10.1109/TCSVT.2012.2221191
36. Zhou, J., Zhou, D., Sun, H., Goto, S.: VLSI architecture of HEVC intra prediction for 8K UHDTV applications. In: IEEE International Conference on Image Processing, pp. 1273–1277. Paris, France (2014). https://doi.org/10.1109/icip.2014.7025254
37. Piao, Y., Min, J., Chen, J.: Encoder improvement of unified intra prediction. In: "Document JCTVC-C207", Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Oct. (2010). https://phenix.int-evry.fr/jct/doc_end_user/documents/3_Guangzhou/wg11/JCTVC-C207-m18245-v2-JCTVC-C207.zip. Accessed 29 Mar 2019
38. Kahng, A., Kang, S.: Accuracy-configurable adder for approximate arithmetic designs. In: ACM/EDAC/IEEE Annual Design Automation Conference, pp. 820–825. San Francisco, USA (2012). https://doi.org/10.1145/2228360.2228509
39. Camus, V., Schlachter, J., Enz, C.: A low-power carry cut-back approximate adder with fixed-point implementation and floating-point precision. In: ACM/EDAC/IEEE Design Automation Conference, pp. 1–6. Austin, USA (2016). https://doi.org/10.1145/2897937.2897964

40. Zhu, N., Goh, W., Zhang, W., Yeo, K., Kong, Z.: Design of low-power high-speed truncation-error-tolerant adder and its application in digital signal processing. IEEE Trans. Very Large Scale Int. Syst. **18**(8), 1225–1229 (2010). https://doi.org/10.1109/tvlsi.2009.2020591
41. Zhu, N., Goh, W., Wang, G., Yeo, K.: Enhanced low-power high-speed adder for error-tolerant application. In: IEEE International SOC Design Conference, pp. 323–327. Incheon, South Korea (2010). https://doi.org/10.1109/socdc.2010.5682905
42. Shafique, M., Ahmad, W., Hafiz, R., Henkel, J.: A low latency generic accuracy configurable adder. In: ACM/EDAC/IEEE Design Automation Conference, pp. 1–6. San Francisco, USA (2015). https://doi.org/10.1145/2744769.2744778
43. Mahdiani, H.R., Ahmadi, A., Fakhraie, S.M., Lucas, C.: Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications. IEEE Trans. Circuits Syst. I Reg. Pap. **57**(4), 850–862 (2010). https://doi.org/10.1109/tcsi.2009.2027626
44. Desoete, B., De Vos Alexis, A.: A reversible carry-look-ahead adder using control gates. Integr. VLSI J. **33**(1), 89–104 (2002)
45. Banerjee, N., et al.: Novel low-overhead operand isolation techniques for low-power datapath synthesis. In: Computer Design: VLSI in Computers and Processors, 2005. ICCD 2005. Proceedings. 2005 IEEE International Conference on IEEE (2005). https://doi.org/10.1109/iccd.2005.80
46. NanGate FreePDK45 Open Cell Library, Nangate [Online]. http://www.nangate.com/?page_id=2325. Accessed 29 Mar 2019
47. Zhou, D., et al.: 14.7 A 4G pixel/s 8/10b H.265/HEVC video decoder chip for 8K ultra HD applications. In: 2016 IEEE International Solid-State Circuits Conference (ISSCC), IEEE (2016). https://doi.org/10.1109/ISSCC.2016.7418009
48. Chuang, T.-D., et al.: A 59.5 mW scalable/multi-view video decoder chip for quad/3D full HDTV and video streaming applications. In: 2010 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), IEEE (2010). https://doi.org/10.1109/ISSCC.2010.5433908
49. Huang, C.-T., et al.: A 249 M pixel/s HEVC video-decoder chip for Quad Full HD applications. In: 2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), IEEE (2013). https://doi.org/10.1109/ISSCC.2013.6487682
50. Tsai, C.-H., et al.: A 446.6 K-gates 0.55–1.2 V H. 265/HEVC decoder for next generation video applications. In: 2013 IEEE Asian Solid-State Circuits Conference (A-SSCC), IEEE (2013). https://doi.org/10.1109/ASSCC.2013.6691043
51. Ju, C.-C., et al.: A 0.2 nJ/pixel 4K 60 fps Main-10 HEVC decoder with multi-format capabilities for UHD-TV applications. In: ESSCIRC 2014-40th European Solid State Circuits Conference (ESSCIRC), IEEE (2014). https://doi.org/10.1109/esscirc.2014.6942055
52. Fang, H., Chen, H., Chang, T.: Fast intra prediction algorithm and design for high efficiency video coding. In: IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1770–1773. Montreal, Canada (2016). https://doi.org/10.1109/iscas.2016.7538911
53. Lu, W., Yu, N., Nan, J., Wang, D.: A hardware structure of HEVC intra prediction. In: 2015 2nd International Conference on Information Science and Control Engineering, pp. 555–559. Shanghai, China (2015). https://doi.org/10.1109/icisce.2015.129
54. Liu, Z., Wang, D., Zhu, H., Huang, X.: 41.7 BN-pixels/s reconfigurable intra prediction architecture for HEVC 2560 × 1600 encoder. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 2634–2638. Vancouver, Canada (2013). https://doi.org/10.1109/icassp.2013.6638133
55. Khan, M., Shafique, M., Grellert, M., Henkel, J.: Hardware-software collaborative complexity reduction scheme for the emerging HEVC intra encoder. In: Proceedings of the conference on design, automation and test in Europe, pp. 125–128. Grenoble, France (2013). https://doi.org/10.7873/date.2013.039
56. Li, F., Shi, G., Wu, F.: An efficient VLSI architecture for 4 × 4 intra prediction in the High Efficiency Video Coding (HEVC) standard. In: 2011 18th IEEE International Conference on Image Processing, pp. 373–376. Brussels, Belgium (2011). https://doi.org/10.1109/icip.2011.6116526
57. Vanne, J., et al.: A high-performance sum of absolute difference implementation for motion estimation. IEEE Trans. Circuits Syst. Video Technol. **16**(7), 876–883 (2006). https://doi.org/10.1109/TCSVT.2006.877150
58. Yufei, L., Xiubo, F., Qin, W.: A high-performance low cost SAD architecture for video coding. IEEE Trans. Consum. Electron. **53**(2), 535–541 (2007). https://doi.org/10.1109/TCE.2007.381726
59. Liu, Z., et al.: Hardware-efficient propagate partial sad architecture for variable block size motion estimation in H. 264/AVC. In: Proceedings of the 17th ACM Great Lakes symposium on VLSI, pp. 160–163. ACM (2007). https://doi.org/10.1145/1228784.1228826

Roger Porto received his M.S. degree in Computer Science from Federal University of Rio Grande do Sul (UFRGS), Brazil, in 2008, and the B.S. degree also in Computer Science from Federal University of Pelotas (UFPel), Brazil, in 2003. He is Professor since 2010 at Sul-rio-grandense Federal Institute of Science, Education and Technology (IFSul), Brazil, and a member of Group of Architectures and Integrated Circuits (GACI) and Video Technology Research Group (ViTech) at UFPel. His research interests include hardware design for video coding and data compression.



Marcel Corrêa received his B.S and M.S. degrees in Computer Science from the Federal University of Pelotas (UFPel), Pelotas-Brazil, in 2013 and 2017 respectively. Currently, he is a Professor at the Sul-rio-grandense Federal Institute of Science, Education and Technology (IFSul) Bagé-Brazil, and a member of the Video Technology Research Group (ViTech). His topics of interest are video coding, data compression and hardware design for video coding and data compression.

Jones Goebel received the B.S. degree in Computer Engineering from the Federal University of Pelotas (UFPel), Pelotas, Brazil, in 2017. He is pursuing the M.Sc. degree in Computer Science from Federal University of Pelotas, Pelotas, RS, Brazil. He is member of the Group of Architectures and Integrated Circuits (GACI) and the Video Technology Research Group (ViTech) from the Federal University of Pelotas. His research interests include image and video coding, digital systems, low-power design and VLSI design for video coding.

Bruno Zatt (M'08–SM'16) received his B.S. and M.S. in Computer Engineering from the Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Brazil, in 2006 and 2008, respectively. He has received his Ph.D. degree in Microelectronics from the same university in 2012 with "summa cum laude" distinction. Currently, Bruno Zatt is a Professor at the Federal University of Pelotas (UFPel), Pelotas, Brazil, and a member of the Group of Architectures and Integrated Circuits (GACI) and the Video Technology Research Group (ViTech). He has 14+ years research experience on algorithms and hardware architectures for video processing including 3 years as an intern researcher at the Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, and experience as a visiting professor at the University of California, Irvine, USA. He has published over 90 papers in international journals/conferences and one book named "3D video coding for embedded devices". Since 2017, Prof. Zatt holds the status of CNPq productivity research fellow.

Nuno Roma (S'01–A'06–M'09–SM'13) received the Ph.D. degree in electrical and computer engineering from Instituto Superior Técnico (IST), Universidade Técnica de Lisboa, Portugal, in 2008. He is an Assistant Professor with the Department of Electrical and Computer Engineering of IST and a Senior Researcher of Instituto de Engenharia de Sistemas e Computadores R&D (INESC-ID). His research interests include computer architectures, specialized and dedicated structures for digital signal processing (including image and video coding), energy-aware computing, parallel processing and high-performance computing systems. He contributed to more than 100 manuscripts to journals and international conferences and served as a Guest Editor of Springer Journal of Real-Time Image Processing (JRTIP) and of EURASIP Journal on Embedded Systems (JES). Dr. Roma is a Senior Member of the IEEE Circuits and Systems Society and a member of ACM.

Luciano Agostini (M'06–SM'11) received the M.S. and Ph.D. degrees in Computer Science from Federal University of Rio Grande do Sul (UFRGS), Brazil, in 2002 and 2007 respectively. He is a Professor since 2002 at Federal University of Pelotas (UFPel), Brazil, where he leads the Video Technology Research Group (ViTech) and the Group of Architectures and Integrated Circuits (GACI). He is advisor at the UFPel Master and Doctorate in Computer Science courses. He was the Executive Vice President for Research and Graduate Studies of UFPel from 2013 and 2017. He has more than 200 published papers in respected international journals and conferences. His research interests include 2D and 3D video coding, algorithmic optimization, arithmetic circuits, and dedicated hardware design. Dr. Agostini is a Senior Member of IEEE and ACM, and a Member of SBC and SBMicro Brazilian societies. He is also a member of the IEEE CAS Multimedia Systems and Applications Technical Committee (MSATC). He is a Brazilian Distinguished Researcher through a CNPq PQ-2 grant.

Marcelo Porto (M'08-SM'17) received the B.S. degree in computer science from the Federal University of Pelotas (UFPel), RS, Brazil, in 2006 and the M.S. and Ph.D. degrees, also in computer science, from the Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, RS, Brazil, in 2008 and 2012, respectively. He is Professor since 2009 at the Center of Technological Development (CDTec) of Federal University of Pelotas, Pelotas, Brazil, member of the Group of Architectures and Integrated Circuits (GACI) and the Video Technology Research Group (ViTech). His research interests include video coding, motion estimation algorithms, FPGA-based design and VLSI design for video coding. He is senior member of IEEE and also member of the IEEE Circuits and System and of the IEEE Signal Processing Society.