VNU Journal of Science:
Computer Science and Communication Engineering

Journal homepage: http://www.jcsce.vnu.edu.vn/index.php/jcsce

Original Article

A Survey of High-Efficient CABAC Hardware Implementations in HEVC Standard

Dinh-Lam Tran, Viet-Huong Pham, Hung K. Nguyen, Xuan-Tu Tran*

VNU Key Laboratory for Smart Integrated Systems (SISLAB),
VNU University of Engineering and Technology, 144 Xuan Thuy road, Cau Giay district, Hanoi, Vietnam

Received 18 April 2019
Revised 07 July 2019; Accepted 20 August 2019

Abstract: High-Efficiency Video Coding (HEVC), also known as H.265 and MPEG-H Part 2, is the newest video coding standard developed to address the increasing demand for higher resolutions and frame rates. In comparison to its predecessor H.264/AVC, HEVC achieved almost double of compression performance that is capable to process high quality video sequences (UHD 4K, 8K; high frame rates) in a wide range of applications. CABAC is the only entropy coding method in HEVC, whose principal algorithm is inherited from its predecessor. However, several aspects of the method that exploits it in HEVC are different, thus HEVC CABAC supports better coding efficiency. Effectively, pipeline and parallelism in CABAC hardware architectures are prospective methods in the implementation of high performance CABAC designs. However, high data dependence and serial nature of bin-to-bin processing in CABAC algorithm pose many challenges for hardware designers. This paper provides an overview of CABAC hardware implementations for HEVC targeting high quality, low power video applications, addresses challenges of exploiting it in different application scenarios and then recommends several predictive research trends in the future.

Keywords: HEVC, CABAC, hardware implementation, high throughput, power saving.

1. Introduction

ITU-T/VCEG and ISO/IEC-MPEG are the two main dominated international organizations that have developed video coding standards [1]. The ITU-T produced H.261 and H.263 while the ISO/IEC produced MPEG-1 and MPEG-4 Visual; then these two organizations jointly produced the H.262/MPEG-2 Video and H.264/MPEG-4 Advanced Video Coding (AVC) standards. The two jointly-developed standards have had a particularly strong impact and have found their ways into a wide variety of products that are increasingly prevalent in

[*] Corresponding author.
E-mail address: tutx@vnu.edu.vn

https://doi.org/10.25073/2588-1086/vnucsce.233

our daily lives. As the diversity of services, the popularity of HD and beyond HD video formats (e.g., 4k×2k or 8k×4k resolutions) have been an emerging trend, it is necessary to have higher coding efficiency than that of H.264/MPEG-4 AVC. This resulted in the newest video coding standard called High Efficiency Video Coding (H.265/HEVC) that developed by Joint Collaborative Team on Video Coding (JCT-VC) [2]. HEVC standard has been designed to achieve multiple goals, including coding efficiency, ease of transport system integration, and data loss resilience. The new video coding standard offers a much more efficient level of compression than its predecessor H.264, and is particularly suited to higher-resolution video streams, where bandwidth savings of HEVC are about 50% [3], [4]. Besides maintaining coding efficiency, processing speed, power consumption and area cost also need to be considered in the development of HEVC to meet the demands for higher resolution, higher frame rates, and battery-based applications.

Context Adaptive Binary Arithmetic Coding (CABAC), which is one of the entropy coding methods in H.264/AVC, is the only form of entropy coding exploited in HEVC [7]. Compared to other forms of entropy coding, such as context adaptive variable length coding (CAVLC), HEVC CABAC provides considerable higher coding gain. However, due to several tight feedback loops in its architecture, CABAC becomes a well-known throughput bottle-neck in HEVC architecture as it is difficult for paralleling and pipelining. In addition, this also leads to high computation and hardware complexity during the development of CABAC architectures for targeted HEVC applications. Since the standard published, numerous worldwide researches have been conducted to propose hardware architectures for HEVC CABAC that trade off multi goals including coding efficiency, high throughput performance, hardware resource, and low power consumption.

This paper provides an overview of HEVC CABAC, the state-of-the-art works relating to the development of high-efficient hardware implementations which provide high throughput performance and low power consumption. Moreover, the key techniques and corresponding design strategies used in CABAC implementation are summarized to achieve the above objectives.

Following this introductory section, the remaining part of this paper is organized as follows: Section 2 is a brief introduction of HEVC standard, CABAC principle and its general architecture. Section 3 reviews state-of-the-art CABAC hardware architecture designs and detailed assess these works in different aspects. Section 4 presents the evaluation and prediction of forthcoming research trends in CABAC implementation. Some conclusions and remarks are given in Section 5.

2. Back Ground of HEVC and CABAC

2.1. HEVC - coding principle and architecture, enhanced features and supported tools

2.1.1. HEVC coding principle

As a successor of H.264/AVC in the development process of video coding standardization, HEVC's video coding layer design is based on conventional block-based hybrid video coding concepts, but with some important differences compared to prior standards [3]. These differences are the method of partition image pixels into Basic Processing Unit, more prediction block partitions, more intra-prediction mode, additional SAO filter and additional high-performance supported coding Tools (Tile, WPP). The block diagram of HEVC architecture is shown in Figure 1.

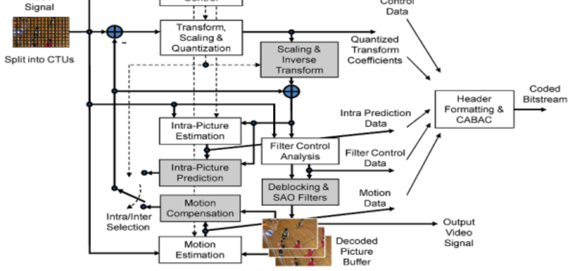


Figure 1. General architecture of HEVC encoder [1].

The process of HEVC encoding to generate compliant bit-stream is typical as follows:

- Each incoming frame is partitioned into squared blocks of pixels ranging from 64×64 to 8×8. While coding blocks of the first picture in a video sequ0065nce (and of the first picture at each clean random-access point into a video sequence) are intra-prediction coded (i.e., the spatial correlations of adjacent blocks), all remaining pictures of the sequence or between random-access points, inter-prediction coding modes (the temporally correlations of blocks between frames) are typically used for most blocks. The residual data of inter-prediction coding mode is generated by selecting of reference pictures and motion vectors (MV) to be applied for predicting samples of each block. By applying intra- and inter- predictions, the residual data (i.e., the differences between the original block and its prediction) is transformed by a linear spatial transform, which will produce transform coefficients. Then these coefficients are scaled, quantized and entropy coded to produce coded bit strings. These coded bit strings together with prediction information are packed and transmitted as a bit-stream format.

- In HEVC architecture, the block-wise processes and quantization are main causes of artifacts of reconstructed samples. Then the two loop filters are applied to alleviate the impact of these artifacts on the reference data for better predictions.

- The final picture representation (that is a duplicate of the output of the decoder) is stored in a decoded picture buffer to be used for the predictions of subsequent pictures.

Because HEVC encoding architecture consists of the identical decoding processes to reconstruct the reference data for prediction and the residual data along with its prediction information are transmitted to the decoding side, then the generated prediction versions of the encoder and decoder are identical.

2.1.2. Enhancement features and supported tools

a. Basic processing unit

Instead of Macro-block (16×16 pixels) in H.264/AVC, the core coding unit in HEVC standard is Coding Tree Unit (CTU) with a maximum size up to 64×64 pixels. However, the size of CTU is varied and selected by the encoder, resulting in better efficiency for encoding higher resolution video formats. Each CTU consists of Coding Tree Blocks (CTBs), in which each of them includes luma, chroma Coding Blocks (CBs) and associated syntaxes. Each CTB, whose size is variable, is partitioned into CUs which consists of Luma CB and Chroma CBs. In addition, the Coding Tree Structure is also partitioned into Prediction Units (PUs) and Transform Units (TUs). An example of block partitioning of video data is depicted in Figure 2. An image is partitioned into rows of CTUs of 64×64 pixels which are further partitioned into CUs of different sizes (8×8 to 32×32). The size of CUs depends on the detailed level of the image [5].

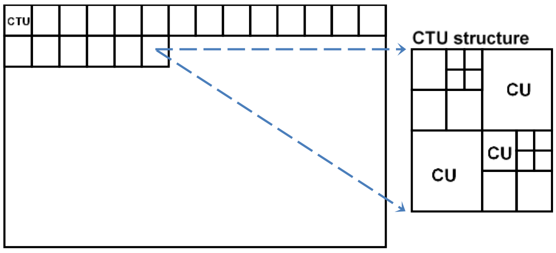


Figure 2. Example of CTU structure in HEVC.

b. Inter-prediction

The major changes in the inter prediction of the HEVC compared with H.264/AVC are in prediction block (PB) partitioning and fractional sample interpolation. HEVC supports more PB partition shapes for inter picture-predicted CBs as shown in Figure 3 [6].

In Figure 3, the partitioning modes of PART−2N×2N, PART−2N×N, and PART−N×2N (with M=N/2) indicate the cases when the CB is not split, split into two equal-size PBs horizontally, and split into two equal-size PBs vertically, respectively. PART−N×N specifies that the CB is split into four equal-sizes PBs, but this mode is only supported when the CB size is equal to the smallest allowed CB size.

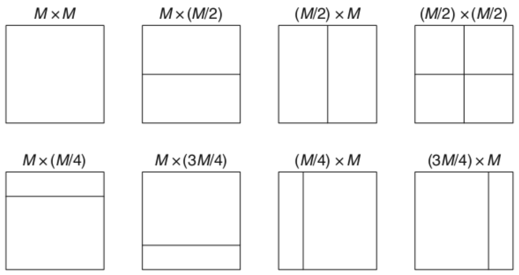


Figure 3. Symmetric and asymmetric of prediction block partitioning.

Besides that, PBs in HEVC could be the asymmetric motion partitions (AMPs), in which each CB is split into two different-sized PBs such as PART−2N×nU, PART−2N×nD, PART−nL×2N, and PART−nR×2N [1], [7]. The flexible splitting of PBs makes HEVC able to support higher compression performance compared to H.264/AVC.

c. Intra-prediction

HEVC uses block-based intra-prediction to take advantage of spatial correlation within a picture and it follows the basic idea of angular intra-prediction. However, HEVC has 35 Luma intra-prediction modes compared with 9 in H.264/AVC, thus provide more flexibility and coding efficiency than its predecessor [7], see Figure 4.

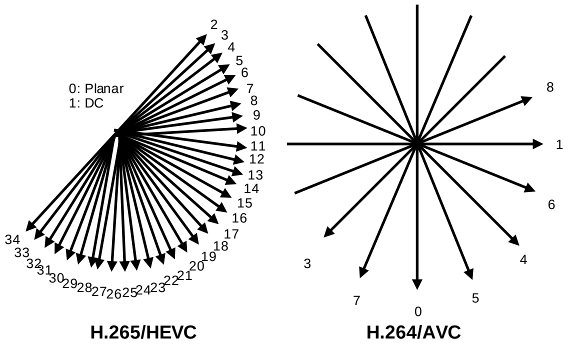


Figure 4. Comparison of Intra prediction in HEVC and H.264/AVC [7].

d. Sample Adaptive Offset filter

SAO (Sample Adaptive Offset) filter is the new coding tool of the HEVC in comparison with H.264/AVC. Unlike the De-blocking filter that removes artifacts based on block boundaries, SAO mitigates artifacts of samples due to transformation and quantization operations. This tool supports a better quality of reconstructed pictures, hence providing higher compression performance [7].

e. Tile and Wave-front Parallel Processing

Tile is the ability to split a picture into rectangular regions that helps increasing the capability of parallel processing as shown in Figure 5 [5]. This is because tiles are encoded with some shared header information and they are decoded independently. Each tile consists of an integer number of CTUs. The CTUs are processed in a raster scan order within each tile, and the tiles themselves are processed in the same way. Prediction based on neighboring tiles is disabled, thus the processing of each tile is independent [5], [7].

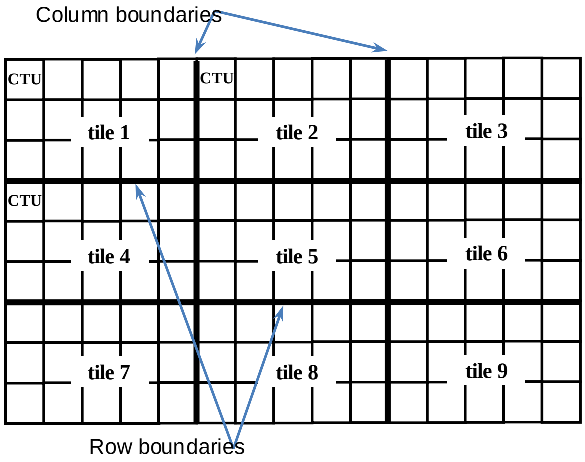


Figure 5: Tiles in HEVC frame [5].

Wave-front Parallel Processing (WPP) is a tool that allows re-initializing CABAC at the beginning of each line of CTUs. To increase the adaptability of CABAC to the content of the video frame, the coder is initialized once the statistics from the decoding of the second CTU in the previous row are available. Re-initialization of the coder at the start of each row makes it possible to begin decoding a row before the processing of the preceding row has been completed. The ability to start coding a row of CTUs before completing the previous one will enhance CABAC coding efficiency.

As illustrated in Figure 6, a picture is processed by a four-thread scheme which speeds up the encoding time for high throughput implementation. To maintain coding dependencies required for each CTU such as each one can be encoded correctly once the left, top-left, top and top-right are already encoded, CABAC should start encoding CTUs at the current row after at least two CTUs of the previous row finish.

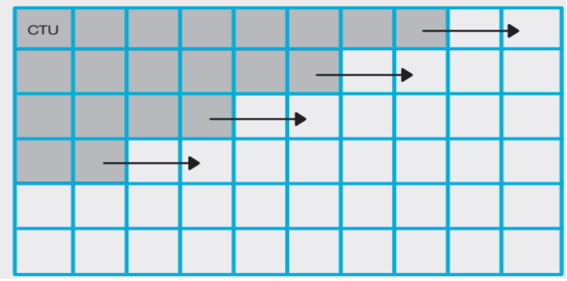


Figure 6. Representation of WPP to enhance coding efficiency.

2.2. CABAC for HEVC (principle, architecture) and its differences from CABAC for H.264

2.2.1. CABAC principle and architecture

While the H.264/AVC uses two entropy coding methods (CABAC and CALVC), HEVC specifies only CABAC entropy coding method.

Figure 7 describes the block diagram of HEVC CABAC encoder. The principal algorithm of CABAC has remained the same as in its predecessor; however, the method used to exploit it in HEVC has different aspects (will be discussed below). As a result, HEVC CABAC supports a higher throughput than that of H.264/AVC, particularly the coding efficiency enhancement and parallel processing capability [1], [8], [9]. This will alleviate the throughput bottleneck existing in H.264/AVC, therefore HEVC becomes the newest video coding standard that can be applied for high resolution video formats (4K and beyond) and real-time video transmission applications. Here are several important improvements according to Binarization, Context Selection and Binary Arithmetic Encoding [8].

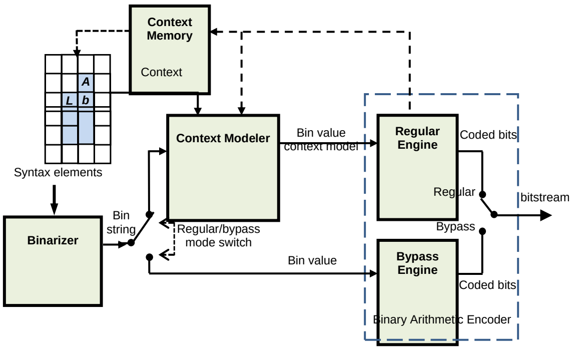


Figure 7. CABAC encoder block diagram [6].

Binarization: This is a process of mapping Syntax elements into binary symbols (bins). Various binarization forms such as Exp-Golomb, fixed length, truncated unary and custom are used in HEVC. The combinations of different binarizations are also allowed where the prefix and suffix are binarized differently such as truncated rice (truncated unary - fixed length combination) or truncated unary - Exp-Golomb combination [7].

Context Selection: The context modeling and selection are used to accurately model the probability of each bin. The probability of bins depends on the type of syntax elements it belongs to, the bin index within the syntax elements (e.g., most significant bin or least significant bin) and the properties of spatially neighboring coding units. HEVC utilizes several hundred different context models, thus it is necessary to have a big Finite State Machine (FSM) for accurately context selection of each Bin. In addition, the estimated probability of the selected context model is updated after each bin is encoded or decoded [7].

Binary Arithmetic Encoding (BAE): BAE will compress Bins into bits (i.e., multiple bins can be represented by a single bit); this allows syntax elements to be represented by a fractional number of bits, which improves coding efficiency. In order to generate bit-streams from Bins, BAE involves several processes such as recursive sub-interval division, range and offset updates. The encoded bits represent an offset that, when converted to a binary fraction, selects one of the two sub-intervals, which indicates the value of the decoded bin. After every decoded bin, the range is updated to equal the selected sub-interval, and the interval division process repeats itself. In order to effectively compress the bins to bits, the probability of the bins must be accurately estimated [7].

2.2.2. General CABAC hardware architecture

CABAC algorithm includes three main functional blocks: Binarizer, Context Modeler, and Arithmetic Encoder (Figure 8). However, different hardware architectures of CABAC can be found in [10], [11], [12], [13], [14].

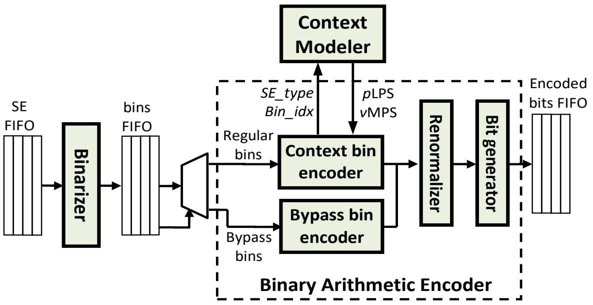


Figure 8. General hardware architecture of CABAC encoder [10].

Besides the three main blocks above, it also comprises several other functional modules such as buffers (FIFOs), data router (Multiplexer and De-multiplexer). Syntax Elements (SE) from the other processes in HEVC architecture (Residual Coefficients, SAO parameters, Prediction mode…) have to be buffered at the input of CABAC encoder before feeding the Binarizer. In CABAC, the general hardware architecture of Binarizer can be characterized in Figure 9.

Based on SE value and type, the Analyzer & Controller will select an appropriate binarization process, which will produce bin string and bin length, accordingly. HEVC standard defines several basic binarization processes such as FL (Fixed Length), TU (Truncated Unary), TR (Truncated Rice), and EGk (kth order Exponential Golomb) for almost SEs. Some other SEs such as CALR (*Coeff_Abs_Level_Remaining*) and QP_Delta (*cu_qp_delta_abs*) utilize two or more combinations (Prefix and Suffix) of these basic binarization processes [15], [16]. There are also simplified custom binarization formats that are mainly based on LUT, for other SEs like Inter Pred Mode, Intra Pred Mode, and Part Mode.

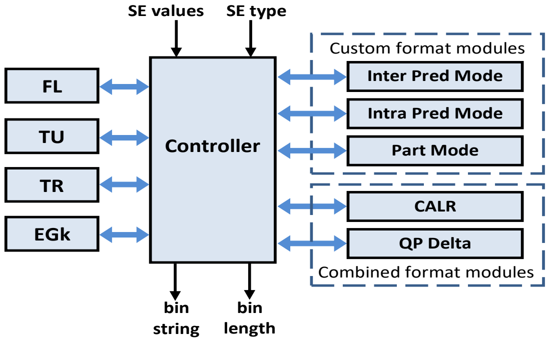


Figure 9. General hardware architecture of a binarizer.

These output bin strings and their bin lengths are temporarily stored at bins FIFO. Depending on bin types (Regular bins or Bypass Bins), the De-multiplexer will separate and route them to context bin encoder or bypass bin encoder. While bypass bins are encoded in a simpler manner, which will not necessary to estimate their probability, regular bins need to be determined their appropriate probably models for encoding. These output bins are put into Bit Generator to form output bit-stream of the encoder.

The general hardware architecture of CABAC context modeler is illustrated in Figure 10. At the beginning of each coding process, it is necessary to initialize the context for CABAC according to its standard specifications, when context table is loaded data from ROM. Depending on Syntax Element data, bin-string from binarizer and neighbor data, the controller will calculate the appropriate address to access and load the corresponding probability model from Context Memory for encoding the current bin. Once the encoding process of the current bin is completed, the context model is updated and written back to Context RAM for encoding the next Bin.

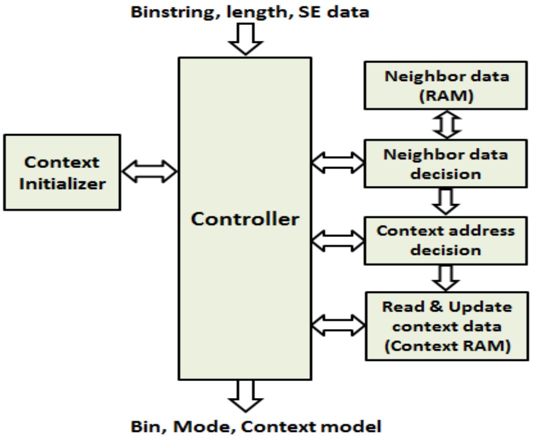


Figure 10. General hardware architecture of context modeller [7].

Binary Arithmetic Encoder (BAE) is the last process in CABAC architecture which will generate encoded bit based on input bin from Binarizer and corresponding probability model from Context Modeler. As illustrated in Figure 8 (CABAC architecture), depending on bin type (bypass or regular), the current bin will be routed into bypass coded engine or context coded engine. The first coded engine is implemented much simpler without context selection and range updating. The coding algorithm of the later one is depicted in Figure 11.

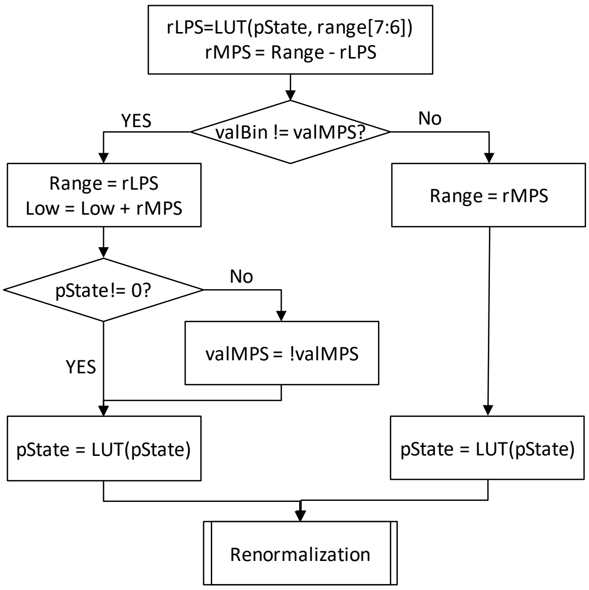


Figure 11. Encoding algorithm of regular coded bin (recommended by ITU-T).

Figure 12 presents our proposed BAE architecture with multiple bypass bin processing to improve the efficiency. The process of BAE can be divided into 4 stages: Sub-intervals division (stage 1 - Packet information extraction and *rLPS* look-up), Range updating (stage 2 - *Range* renormalization and pre-multiple bypass bin multiplication), Low updating (stage 3 - *Low* renormalization and outstanding bit look-up), and Bits output (stage 4 - Coded bit construction and calculation of the number of valid coded bits). The inputs to our architecture are encapsulated into packets in order to enable multiple-bypass-bin processing. Each packet could be a regular or terminate bin or even a group of bypass bins. The detailed implementation of these stages can be found in our previous work [17].

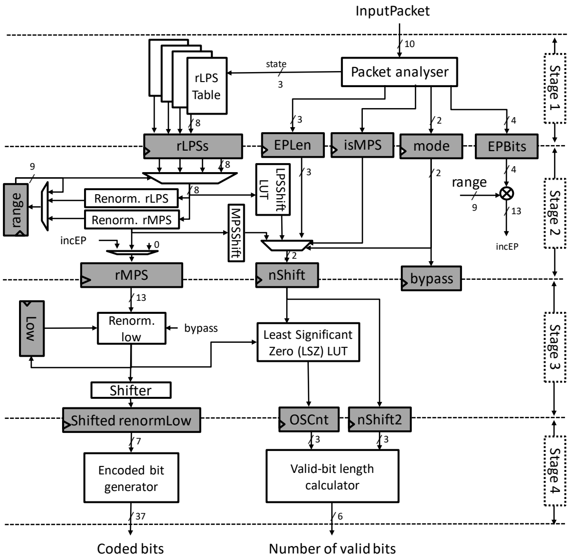


Figure 12. Hardware implementation of regular bin encoding [17].

2.2.3. Differences between HEVC CABAC and H.264/AVC CABAC

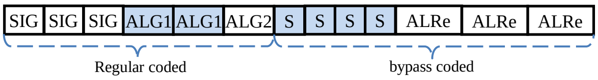
In terms of CABAC algorithm, Binary arithmetic coding in HEVC is the same with H.264, which is based on recursive sub-interval division to generate output coded bits for input bins [7]. However, because HEVC exploits several new coding tools and throughput improvement oriented-techniques, statistics of bins types are significantly changed compared to H.264 as shown in Table 1.

Table 1. Statistics of bin types in HEVC and H.264/AVC standards [8]

	Common condition configuration	Context (%)	Bypass (%)	Terminate (%)
H.264/AVC	Hierarchical B	80.5	13.6	5.9
	Hierarchical P	79.4	12.2	8.4
HEVC	Intra	67.9	32.0	0.1
	Low delay P	7.2	20.8	1.0
	Low delay B	78.2	20.8	1.0
	Random access	73.0	26.4	0.6

Obviously, in most condition configurations, HEVC shows a fewer portion of Context coded bin and Termination Bins, whereas Bypass bins occupy considerably portion in the total number of input bins.

HEVC also uses less number of Contexts (154) than that of H.264/AVC (441) [1], [8]; hence HEVC consumes less memory for context storage than H.264/AVC that leads to better hardware cost. Coefficient level syntax elements that represent residual data occupies up to 25% of total bins in CABAC. While H.264/AVC utilizes TRU+EGk binarization method for this type of Syntax Element, HEVC uses TrU+FL (Truncated Rice) which generates fewer bins (53 vs. 15) [7], [8]. This will alleviate the workload for Binary arithmetic encoding which contributes to enhance the CABAC throughput performance. The method of characterizing syntax elements for coefficient levels in HEVC is also different from H.264/AVC which lead to possibility to group the same context coded bins and group bypass bins together for throughput enhancement as illustrated in Figure 13 [8].



SIG	SIG	SIG	ALG1	ALG1	ALG2	S	S	S	S	ALRe	ALRe	ALRe

Figure 13. Group same regular bins and bypass bins to increase throughput.

Table 2. Reduction in workload and memory of HEVC over H.264/AVC [8]

Metric	H.264/AVC	HEVC	Reduction
Max regular coded bins	78825	882	9x
Max bypass bins	13056	13417	1x
Max total bins	20882	14301	1.5x
Number of contexts	441	154	3x
Line buffer for 4K×2K	30720	1024	30x
Coefficient storage	8×8×9-bits	4×4×3-bits	12x
Initialization Table	1746×16-bits	442×8-bits	8x

This arrangement of bins gives better chances to propose parallelized and pipelined

CABAC architectures. Overall differences between HEVC and H.264/AVC in terms of input workload and memory usage are shown in Table 2.

3. HEVC CABAC implementations: State-of-the-Art

3.1. High throughput design strategies

In HEVC, CABAC has been modified all of its components in terms of both algorithms and architectures for throughput improvements. For Binarization and Context Selection processes, there are commonly five techniques to improve the throughput of CABAC in HEVC. These techniques are reducing context code bins, grouping bypass bins together, grouping the same context bins together and reducing the total number of bins [7], [9], [18]. These techniques have strong impacts on architect design strategies of BAE in particular and the whole CABAC as well for throughput improvement targeting 4K, 8K UHD video applications.

a) Reducing the number of context coded bins

HEVC algorithm supports to significantly reduce the number of context coded bins for syntax elements such as motion vectors and coefficient level. The underlying cause of this reduction is the relational proportion of context coded bins and bypass coded bins. While H.264/AVC uses a large amount of context coded bins for syntax elements, HEVC only uses the first few context coded bins and the remaining bins are bypass coded. Table 3 summarizes the reduction in context coded bins for various syntax elements.

Table 3. Comparison of bypass bins number [9]

Syntax element	AVC	HEVC
Motion vector difference	9	2
Coefficient level	14	1 or 2
Reference index	31	2
Delta QP	53	5
Remainder of intra prediction mode	3	0

b) Grouping of bypass bins

Once the number of context coded bin is reduced, bypass bins occupy a significant portion of the total bins in HEVC. Therefore, overall CABAC throughput could be notably improved by applying a technique called “grouping of bypass bins” [9]. The underlying principle is to process multiple bypass bins per cycle. Multiple bypass bins can only be processed in the same cycle if bypass bins appear consecutively in the bin stream [7]. Thus, long runs of bypass bins result in higher throughput than frequent switching between bypass and context coded bins. Table 4 summarizes the syntax elements where bypass grouping was used.

Table 4. Syntax Element for group of bypass bins [9]

Syntax element	**Nbr of SEs**
Motion vector difference	2
Coefficient level	16
Coefficient sign	16
Remainder of intra prediction mode	4

c) Grouping bins with the Same Context

Processing multiple context coded bins in the same cycle is a method to improve CABAC throughput. This often requires speculative calculations for context selection. The amount of speculative computations, which will be the cause for critical path delay, increases if bins using different contexts and context selection logic are interleaved. Thus, to reduce speculative computations hence critical path delay, bins should be reordered such that bins with the same contexts and context selection logic are grouped together so that they are likely to be processed in the same cycle [4], [8], [9]. This also reduces context switching resulting in fewer memory accesses, which also increases throughput and power consumption as well.

d) Reducing the total number of bins

The throughput of CABAC could be enhanced by reducing its workload, i.e. decreasing the total number of bins that it needs to process. For this technique, the total number of bins was reduced by modifying the binarization algorithm of coefficient levels. The coefficient levels account for a significant portion on average 15 to 25% of the total number of bins [18]. In the binarization process, unlike combined TrU + EGk in AVC, HEVC uses combined TrU + FL that produce much smaller number of output bins, especially for coefficient value above 12. As a result, on average the total number of bins was reduced in HEVC by 1.5x compared to AVC [18].

Binary Arithmetic Encoder is considered as the main cause of throughput bottle-neck as it consists of several loops due to data dependencies and critical path delays. Fortunately, by analyzing and exploiting statistical features, serial relations between BAE and other CABAC components to alleviate these dependencies and delays, the throughput performance could be substantially improved [4]. This was the result of a series of modifications in BAE architectures and hardware implementations such as paralleled multiple BAE, pipeline BAE architectures, multiple-bin single BAE core and high speed BAE core [19].

The objective of these solutions is to increase the product of the number of processed bins/clock cycle and the clock speed. In hardware designs for high performance purpose, these criteria (bins/clock and clock speed) should be traded-off for each specific circumstance as example depicted in Figure 14.

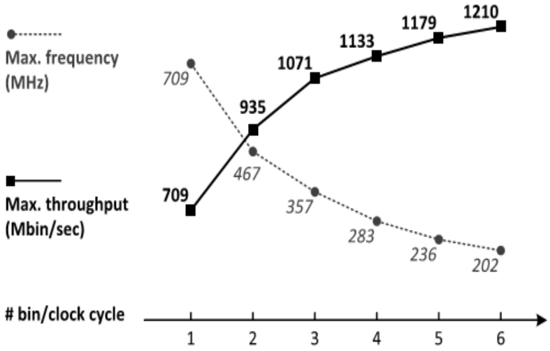


Figure 14. Relationship between throughput, clock frequency and bins/cycle [19].

Over the past five-year period, there has been a significant effort from various research groups worldwide focusing on hardware solutions to improve throughput performance of HEVC CODEC in general and CABAC in particular. Table 5 and Figure 15 show highlighted work in CABAC hardware design for high performance.

Table 5: State-of-the-art high performance CABAC implementations

Works (year)	**[11]** (2013)	**[12]** (2015)	**[13]** (2015)	**[23]** (2017)	**[21]** (2018)	**[20]** (2016)	**[22]** (2016)	**[19]** (2014)	**[14]** (2013)
Bins/clk	1.18	2.37	1	3.99	4.94	4.07	1.99	4.37	4.4
Max Frequency (MHz)	357	380	158	625	537	436.7	1110	420	402
Max throughput (Mbins/s)	439	900	158	2499	2653	1777	2219	1836	1769
Tech.	130	130	180	65	65	40	65	90	65
Area (kGate)	48.94	31.1	45.1	11.2	33	20.39	5.68	111	148
Design strategies	Parallel CM (Whole CABAC design)	Area efficient Multi-bin Binarizer and parallel BAE	Fully CABAC pipelined	Combined Parallel, pipeline in BAE	8-stage pipeline multi bins BAE	High speed multi bin BAE	4-stage pipeline BAE	Combined Parallel, pipeline in both BAE and CM	High speed, Multi-bin pipeline architecture CABAC

Throughput performance and hardware design cost are the two focusing design criteria in the above work achievements. Obviously, they are contrary and have to be trade-off during design for specific applications. The chart shows that some work achieved high throughput with large area cost [14], [19] and vice versa [11], [12], [13]. Some others [20], [21], [22] achieved very high throughput but consumed moderate, even low area. It does not conflict with the above conclusion, because these works only focused on BAE design, thus consuming less area than those focusing on whole CABAC implementation. These designs usually achieve significant throughput improvements because BAE is the most throughput bottle-neck in CABAC algorithm and architecture. Therefore, its improvement has huge effects on the overall design.

Peng *et al.* [11] proposed a CABAC hardware architecture, as shown in Figure 16 which not only supports high throughput by a parallel strategy but also reduce hardware cost. The key techniques and strategies that exploited in this work are based on analyzing statistics and characteristics of residual Syntax Elements (SE). These residual data bins occupy a significant portion in total bins of CABAC, thus an efficient coding method of this type of SE will contribute to the whole CABAC implementation. Authors propose a method of rearranging this SE structure, Context selection and binarization to support parallel architecture and hardware reduction. Firstly, SEs represent residual data [6] (last_significant_coeff_x,last_significant_coeff_y, coeff_abs_ level_greater1_flags, coeff_abs_level_greater2_flag, coeff_ abs_level_remaining and coeff_sign_flag) in a coded sub-block which are grouped by their types as they are independent context selection. Then context coded and bypass coded bins are separated.

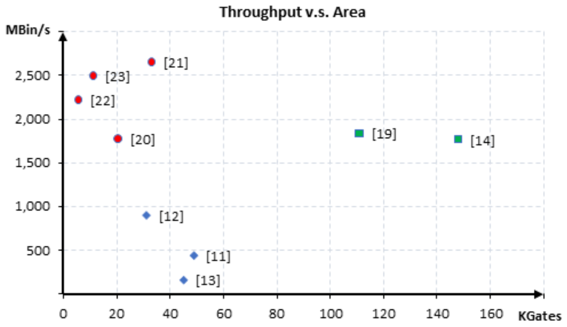


Figure 15. High performance CABAC hardware implementations.

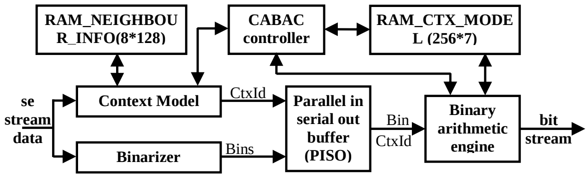


Figure 16. CABAC encoder with proposed parallel CM [11].

The rearranged structure of SEs for residual data is depicted in Figure 17. This proposed technique allows context selections to be paralleled, thus improve context selection throughput 1.3x on average. Because of bypass coded bins are grouped together, they are encoded in parallel that contributes to throughput improvement as well. A PISO (Parallel In Serial Out) buffer is inserted in CABAC architecture to harmonize the processing speed differences between CABAC sub-modules.

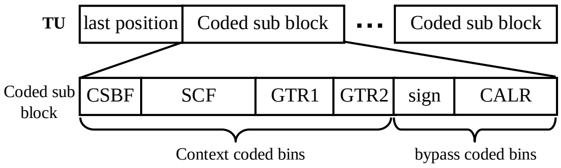


Figure 17. Syntax structure of residual data in CABAC encoder [11].

For hardware cost reduction: the design of a context-based adaptive CALR binarization hardware architecture can save hardware resource while maintaining throughput performance. The bin length is adaptively updated in accordance with cRice Parameter (cRiceParam). The hardware solution for CALR binarization process applied in CABAC design is shown in Figure 18.

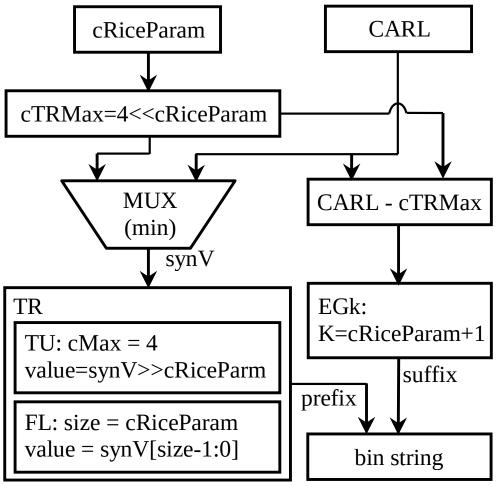


Figure 18. Adaptive binarization implementation of CARL [11].

D.Zhou *et al.* [19] focuses on designing of an ultra-high throughput VLSI CABAC encoder that supports UHDTV applications. By analyzing CABAC algorithms and statistics of data, authors propose and implement in hardware a series of throughput improvement techniques (pre-normalization, Hybrid Path Coverage, Look-ahead rLPS, bypass bin splitting and State Dual Transition). The objectives of the above techniques and design solutions are both critical path delay reduction and the increase of the number of processed bins per clock cycle, thus improving CABAC throughput. To support multi-bin BAE, they proposed a cascaded 4-bin BAE as shown in Figure 19.

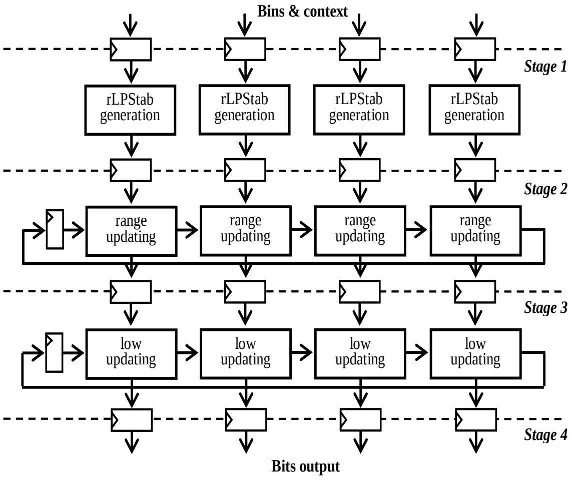
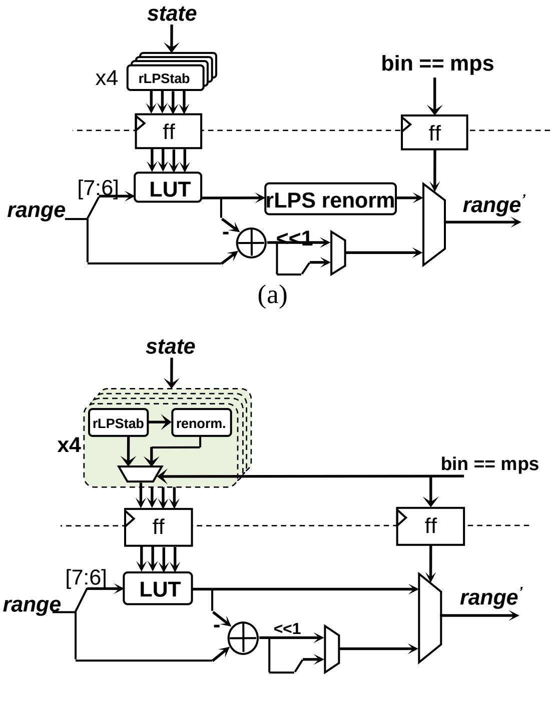


Figure 19. Proposed hardware architecture of cascaded 4-bin BAE [19].

In Figure 19, because of bin-to-bin dependency and critical delay in the stage 2 of BAE process, the cascaded architecture will further expand this delay that degrades clock speed, hence reducing the throughput performance. Two techniques (Pre-norm, HPC) are applied to solve this issue, in which pre-norm will shorten the critical delay of stage 2 and HPC will reduce cascaded 4-bin processing time.

Pre-norm implementation in Figure 20(a) is original stage 2 of BAE architecture, while Figure 20(b) will remove the normalization from the stage 2 to the stage 1, which is much less processing delay.

To further support the cascaded 4-bin BAE architecture, they proposed LH-rLPS to alleviate the critical delay of range updating through this proposed multi-bin architecture. The conventional architecture is illustrated in Figure 21, where the cascaded 2-bin range updating undergoes two LUTs.



(a)

(b)

Figure 20. Conventional renormalization (a); and Pre-normalization technique (b) [19].

Figure 22 shows the proposed architecture, which functionality is equivalent to that of Figure 21, where the range updating goes through only one LUT instead. This is solved by adding a pre-routing 4-4 router at the previous stage to specify 4 potential candidates of the second LUT's inputs. The two upper bits of input range 1 will be utilized to select one out of these 4 candidates as the final updated range value. This cascaded 2-bin range updating architecture can be generalized to design the four-bin architecture as shown in Figure 23, which will be applied for the above proposed four-bin CABAC architecture.

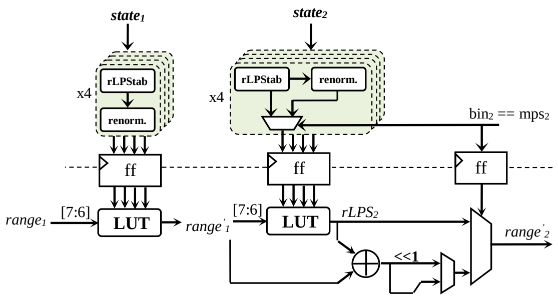


Figure 21. Critical delay in cascaded range-update [19].

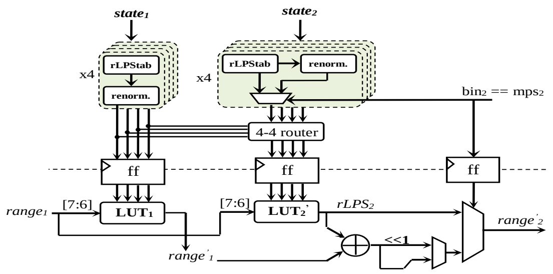


Figure 22. LH-rLPS for cascaded L-F update [19].

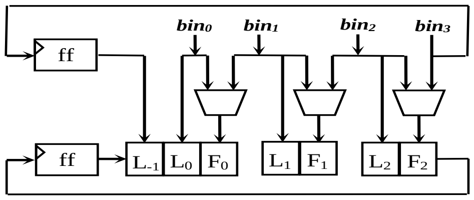


Figure 23. Proposed 4-bin LH-rLPS architecture [19].

In addition to the above three solutions for reducing critical path delay in the proposed four-bin pipeline CABAC architecture, a BPPS technique is applied to improve number of bins processed per clock cycle. In HEVC, bypass bins occupy relatively high proportion up to 30%. Since arithmetic coding for bypass bins do not influence the range updating unit and context model, they can be removed from the range updating stage for multiple bypass bin processing in a clock cycle.

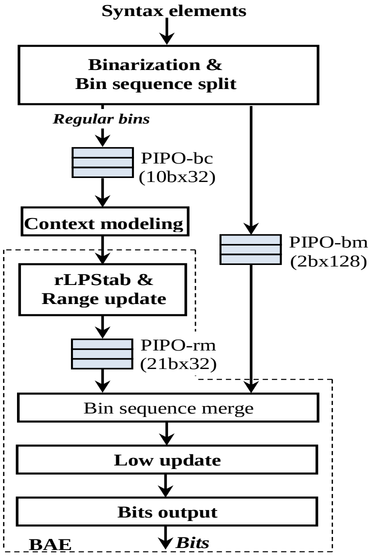


Figure 24. Proposed four-bin BAE with BPBS [19].

A BPBS scheme is proposed to split the bypass bins from the bin stream together with their relative positions are forwarded through a dedicated channel - a PIPO (Parallel In Parallel Out) storage - until re-merged into the bin-stream before the low update stage, as shown in Figure 24. This will result in improving BAE throughput as it is possible to process several bypass bins in one clock cycle as well as several bypass bins in parallel.

The above sub-architectures which are the realizations of each proposed throughput enhancement technique are embedded into the 4-bin pipelined CABAC architecture that supports UHDTV applications.

Considering Binary Arithmetic Encoder (BAE) only, Chen *et al.* [20] also exploited Pre-normalization and Look-ahead rLPS techniques to design high throughput BAE architecture for HEVC CABAC. However, there was a modification in the hardware implementation of renormalization for range updating in comparison with design solution used in [19]. As a result, the processing delay of range updating is further optimized for high speed multi-bin BAE architecture. The proposed hardware modification is illustrated in Figure 25. As shown in this figure, two 2-1 MUXs are removed from the critical range updating stage compared to original architecture in Figure 26. This will lead to a further reduction in processing delay of range updating that contributes to throughput improvement. The architecture is simplified due to the application of PBN (Pre-Bitwise-NOT) operation for range updating, which processes rLPS renormalization with PBN at the first stage of pipeline BAE. This design strategy is then exploited in cascaded 4-bin BAE architecture, where Look-ahead rLPS architecture is further optimized. Even though there were fewer proposed techniques applied in hardware design in comparison with [19], the throughput of proposed BAE architecture in this design almost reaches the achievement of [19]. This is because of optimization in processing delay that leads to significant improvement in operating frequency.

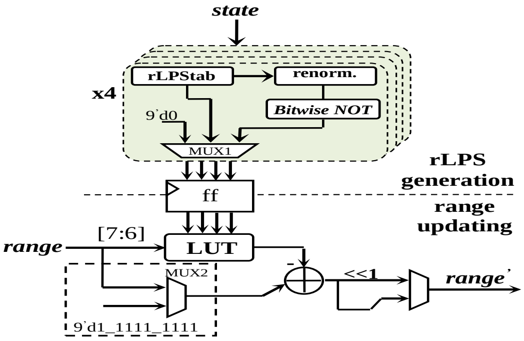


Figure 25. Proposed range updated with PBN [20].

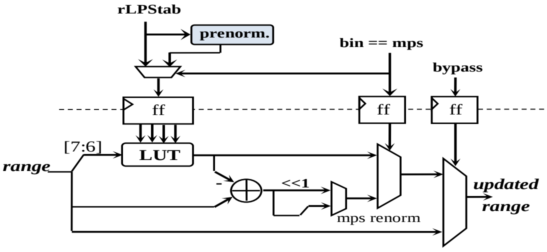


Figure 26. Range-update architecture [20].

D. Kim *et al.* [13] proposed a pipeline HEVC CABAC architecture that support speeding up the operating frequency. It is a highly pipelined CABAC design in all of its components to optimize the processing delay as shown in Figure 27. In this architecture, to overcome differences in processing speed between stages due to pipeline strategies, FIFOs are inserted.

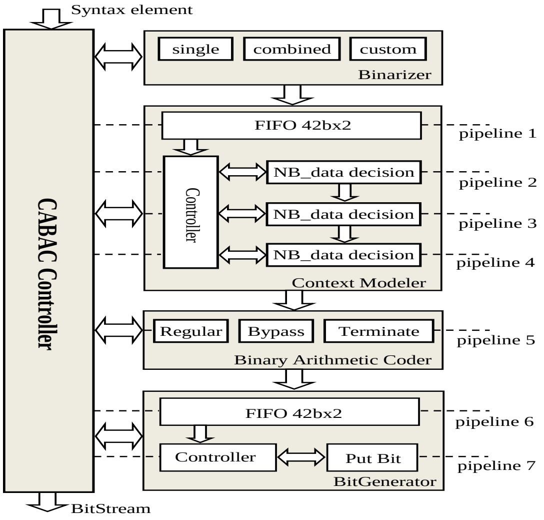


Figure 27. Seven-stage pipeline CABAC hardware architecture [13].

In the proposed CABAC hardware architecture, binarization module is designed as shown in Figure 28, where a multiple and parallel SEs can be processed for the following pipeline stage of the whole architecture. Additionally, Context Modeler is also well-designed for four-stage pipeline architecture that is detailed in Figure 29. Finally, when all of the above stages are pipelined to shorten delay, the critical processing delay occurs in the last stage (Bit Generator) of CABAC. A separation of renormalization and bit output is proposed in this stage to balance the average delay. As a result of this careful separation, a seven-stage pipelined HEVC CABAC is proposed, implemented and successfully tested.

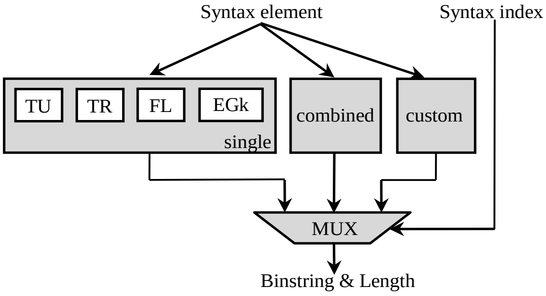


Figure 28. A binarizer for CABAC implementation [13].

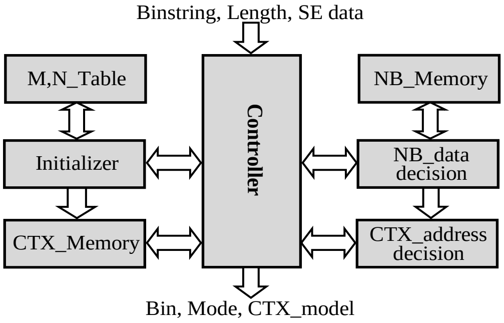


Figure 29. Architecture of four stage pipeline CM [7].

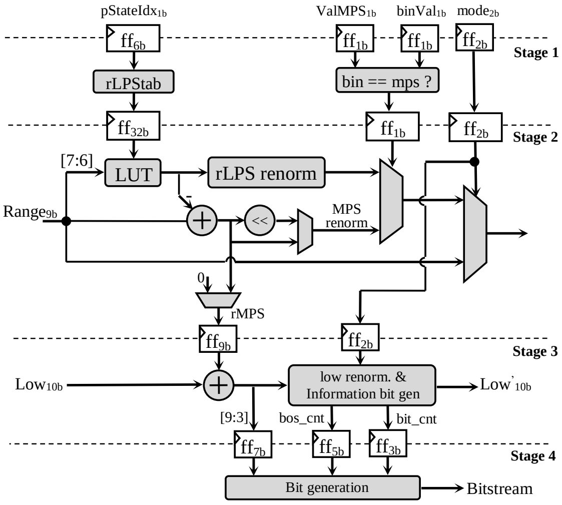


Figure 30. Four-stage pipeline BAE [22].

H. Jo *et al.* [22] proposed an efficient four-stage pipeline BAE that splits several processes in stage 2 to provide high parallelism. Figure 30 shows proposed BAE hardware architecture in a single bin, while a cascaded multi-bin BAE architecture is proposed based on the single bin one as shown in Figure 31.

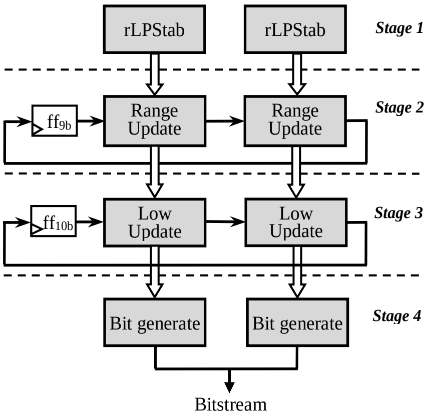


Figure 31. Cascaded two-bin four-stage pipeline BAE architecture [22].

Traditionally, once renormalization is necessary, the process has to be iterated until the range ≥ 256 with no prior knowledge, resulting in long delay. This is indicated by the loop in Figure 32. This variable loop has been solved by pre-calculating renormalization number by finding the first "1" position from MSB of current range value. Then the number of left shifts is determined and accomplished accordingly as shown in Figure 33. This technique optimally reduces critical processing delay that allows a four-stage pipelined BAE architecture working at high speed. This pipelined architecture is also applied in two-bin BAE architecture to improve throughput performance.

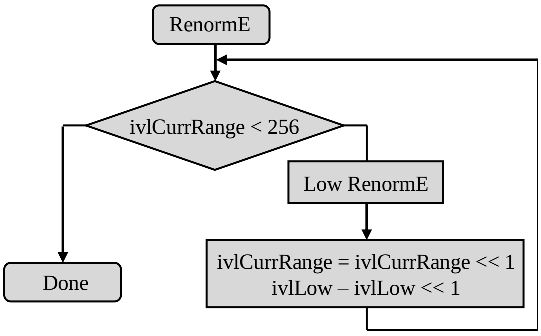


Figure 32. Variable renormalization loop causing delay [22].

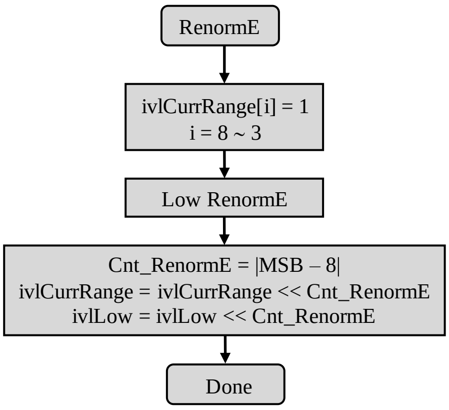


Figure 33. Proposed renormalization technique avoiding loop [22].

In the work done by B. Vizzotto *et al.* [12], for the purpose of encoding UHD video contents, an area efficient and high throughput CABAC encoder architecture is presented. To achieve the desired objectives, two design strategies have been proposed to modify CABAC hardware architecture. Firstly, parallel binarization architecture has been designed to meet the requirement of high throughput as depicted in Figure 34. The proposed architecture supports encoding multi SEs for multi-bin CABAC architecture. However, instead of a parallel binarization for each format which consumes large hardware a heterogeneous eight-functional-core binarizer has been presented to save area cost. This heterogeneous architecture consists of eight cores that can process up to 6 SEs per clock cycle due to the duplication of Custom, TU and EGk cores in the design.

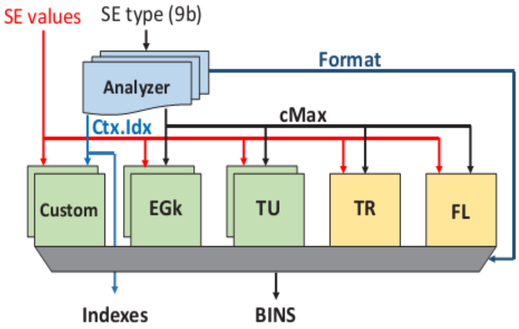


Figure 34. Parallel binarization architecture [12].

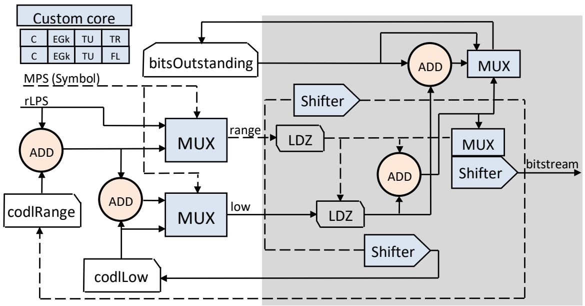


Figure 35. Renormalization architecture in BAE [12].

The second solution focuses on speeding up renormalization process of BAE. Based on the Leading Zero Detector (LDZ) proposal presented in [3] to alternate variable loop in renormalization for shortening critical delay, Vizzotto et al. proposed the renormalization architecture as shown in Figure 35. In this proposed architecture, two LDZs are utilized in parallel for Range and Low renormalization, thus further reduce the processing time of this BAE stage.

Ramos *et al*. [21] proposed 8-stage pipeline BAE architecture which can integrate multi-bypass bin processing (MBBP) to achieve the highest throughput among related works in the literature. Figure 38 shows the proposed architecture that fully integrates high speed techniques (Pre-norm, LH-rLPS, BPBS) in a highly pipelined BAE (8-stage). In addition, an MBBP scheme is proposed to further increase the number of processed bins per clock cycle that contribute to throughput achievement. Once bypass bins are separated from regular bins to avoid critical processing in range update, there are possibilities to process multi-bypass bins in the following stage, i.e. low updating, for throughput improvement. To utilize MBBP in 8-stage pipeline BAE core without degradation of the operating frequency, it is necessary to separate and pipelined process low update for bypass bins as the proposed architecture shown in Figure 36. In this hardware implementation, low update algorithm of bypass bin is separated into two sub-stages to realize two multiply operations, which can balance processing delay to other stages in the pipeline architecture. In addition, these multiple operations are alternated by combinations of adder, shifter and multiplexer to reduce critical delay and area cost.

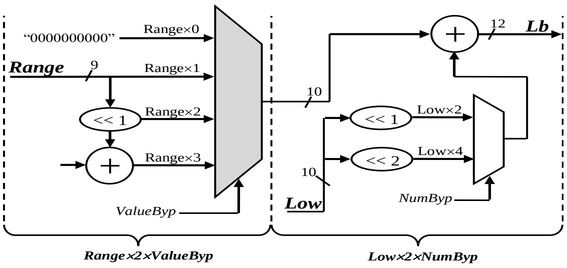


Figure 36. Pipeline low-update architecture for BAE [21].

This split low update architecture is then exploited into fifth and sixth stages of proposed BAE in Figure 37.

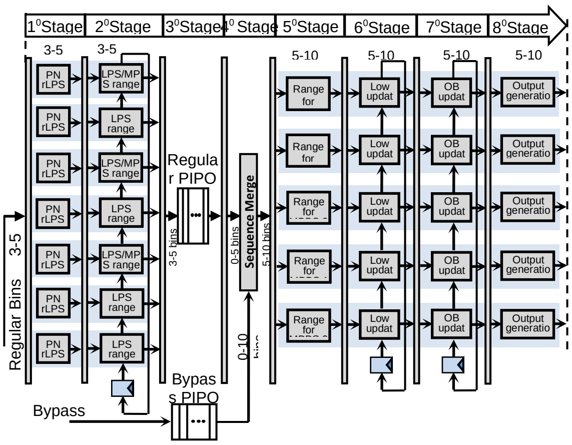


Figure 37. Eight-stage pipeline multi-bypass-bin-scheme BAE architecture [21].

In this 8-stage pipeline BAE architecture, all advanced techniques (PN rLPS, LH rLPS and BPBS) to improve throughput are integrated. The first stage is rLPS pre-selection using PN rLPS technique. The second is regular bin range updating that applies LH rLPS in seven cores scheme, followed by PIPO write/read regular buffer before re-merging to bypass stream at the fourth stage. As mentioned, the next two stages are catered for

Low updating process with a five-core scheme. The seventh stage is five-core OB (Output Bits) that is separated from low updating for reducing process delay purpose. The last stage is for final bit generation.

3.2. Low power design strategies

The application of low-power techniques (clock-gating, power-gating, DVFS) to design HEVC hardware architectures for low power consumption is quite diverse and complicated, especially for CABAC module. Depending on the functional module and the characteristics of processed data the most reasonable technique can be determined [25]. Binarizer is the first module that can be designed with low power consumption based on statistical analyzing of input data (Syntax Elements). The analysis results indicate that FL (Fixed Length) is the most widely used binarization method in HEVC standard. In addition, when this binarization format is activated it tends to be invoked repeatedly to form the same type of bin-string. Then the AND-based Operand Isolation technique was proposed to insert into Binarizer architecture for energy saving purpose [24]. The application of this technique in one-core and four-core binarization architectures are shown in Figure 38 and Figure 39.

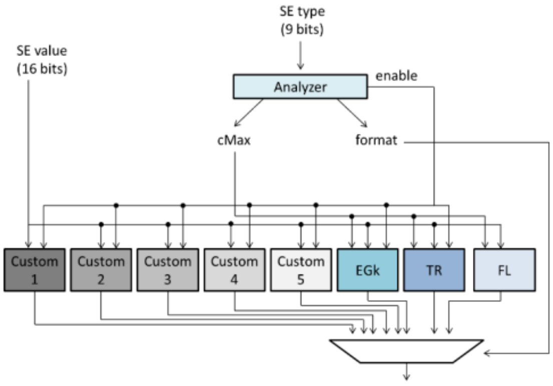


Figure 38. One-core binarization architecture [24].

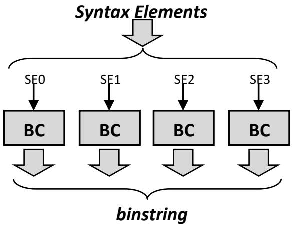


Figure 39. Four-core parallel binarization architecture [24].

For high throughput requirement, four-Binarization Core (BC) architecture is proposed (Figure 39), in which hardware architecture of each BC is shown in Figure 38. Based on statistical analysis concluded above, AND-based Operand Isolation technique is inserted into each BC for low power purpose as shown in Figure 40. Except for FL format that occupies a significant portion in Binarization, the proposed low power technique is embedded into the inputs of all other binarization processes. Then the less frequently used binarization formats will be deactivated and isolated. As a result, the power consumption of binarizer is reduced by 20% on average.

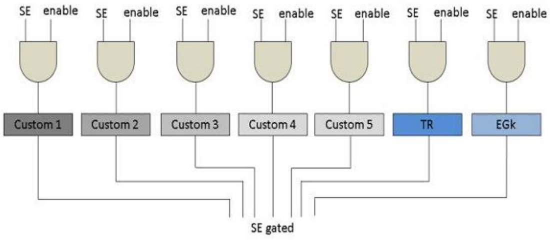


Figure 40. AND-based isolation operand for low-power binarization architecture [24].

BAE is another sub-module in CABAC that can apply low power technique for energy saving in hardware design. Unlike Binarizer that exploits the statistical analysis of SE types to propose appropriate low power solutions, BAE works with bin types (regular, bypass and MPS/LPS). Use the same method, the statistics analysis concluded that the occurrence of regular bins is more frequent than bypass ones and MPS regular bins tend to occur in longer bursts than LPS regular Bins. Therefore, when bypass bins and regular bins are separated by

BPBS technique and bypass bins are grouped together, it is possible to turn off bypass bin processing path while long bursts of regular bins are processed and vice versa. In the regular bin processing path, it is also possible to proposed power saving to turn off LPS bin processing part in BAE architecture. These power saving solutions in BAE architecture can be realized by exploiting Clock Gating technique.

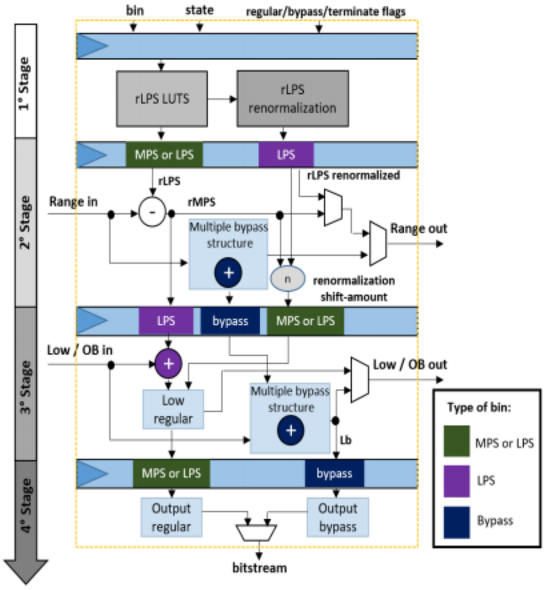


Figure 41. Clock Gating for low-power BAE architecture [26].

Ramos *et al*. [26] proposed a novel Multiple Bypass Bins Processing (MBBP) Low-power BAE architecture that supports 8K UHD video. They proposed a multiple bins architecture that can process multiple bins per clock which can give opportunities to minimize clock frequency while still be able to satisfy 8K UHD video application. The lower frequency can be applied the lesser power consumption can be achieved. Moreover, this is also a four-stage pipeline BAE, in which Clock Gating technique can be applied for stage pipeline registers based on data path for each kind of bins as stated above. These clock-gated pipeline registers will contribute to energy saving by an appropriate control mechanism. As a result, the BAE architecture shown in Figure 41 is capable to process 8k UHD video sequences at the minimum clock frequency that leads to a power saving of 14.26%.

4. HEVC CABAC Research Trends

Since the compression efficiency of HEVC is almost double of that in H.264/AVC, HEVC is considered as the promising standard for video applications today. However, the improvement of compression efficiency comes at the cost of increased computational complexity, large processing delays, higher resource and energy consumption compared to H.264/AVC. These fundamental problems significantly affect the realization of HEVC standards in today's video applications. The emerging challenges are the exploitation of HEVC standard in widespread video applications, where real-time high quality UHD video is transmitted over the limited bandwidth wireless media (broadcast TV, mobile network, satellite communication and TV) and network delivery [27]. In these video services, it is necessary to transmit a large amount of real-time video data over a bandwidth-limiting transmission medium and unstable channel quality that affects video quality. In addition, most of the terminal devices (mobile phones, tablet, camcorders…) in these video transmission systems are resource constrained in terms of storage, computation and processing capacity, energy (battery) and network bandwidth, which are the other hindrance for the realization of HEVC standard [28], [29].

Recently, both academic and industrial sectors have focused on the explorations of solutions for overcoming the above challenges and it can be predicted that these research trends will be continued in the future as long as HEVC standard has been not fully adopted into modern video services [30]. It is obvious that these challenges will be posed to transmission media operators, video service providers as well as terminal holders. Thus, a complete solution is accomplished by contributions from above-named and research trends are categorized accordingly. For the objective of the utilization of HEVC standard in video transmission systems, the ongoing research trends could be divided into the following themes:

- Optimizing algorithms and hardware architectures for HEVC codec to meet the demand of resource-constrained devices.
- Developing adaptive encoding schemes to network conditions when applying in live broadcasting and real-time streaming over wireless networks.
- Proposing reconfigurable video coding (RVC) architectures that are able to effectively adopt HEVC standard into existing systems and infrastructures, where the previous standards have been already integrated.

The first theme is ongoing conventional research direction and it could be considered the underlying foundation that others based on. Mobile video applications have started to dominate the global mobile data traffic in recent years. In most mobile video communication systems, mobile users will be equipped with mobile devices, which are typically resource constrained in terms of storage, computation and processing capacity, energy (battery) and network bandwidth [28], [29]. This issue has been already addressed and drawn a tremendous research since commencing the first version of HEVC standard. As mentioned in previous sections, to support resource-constrained applications, most of the components in HEVC architecture have been assessed and amended to enhance its performance. However, the necessity of this improvement is always at a high demand to better support future video applications. Appling convolution neural network and machine learning to propose adaptation algorithms, e.g., QP adaptation, could be a potential method for this research direction [31], [32]. Additionally, the more flexibly and adaptively encoding schemes for integrating HEVC into existing infrastructure at any transmitting media condition are also promising directions. Thus, there should be adaptive algorithms to estimate the HEVC encoder model parameters and perform online encoder coding configuration optimization [33].

For the second research direction, the advance in coding efficiency of HEVC standard makes it become a candidate in limited bandwidth video communication system, particularly in wireless media such as TV broadcasting, satellite communication and mobile networks. In these wireless media, the quality of service depends not only on channel bandwidth but also heavily the variation in environmental conditions [29]. The challenge arises in the application scenarios of live broadcasting and real-time streaming over the wireless networks. These services impose a very high rate, large amounts of data traversing through the networks. The feasible solution for this challenge is to dynamically adapt encoding schemes to network conditions for a better trade-off between quality of services, efficiency in exploiting encoding and network capabilities. Rate control has always been a potential research area that supports dynamically adapt encoding solution in wireless live video services. However, there has been insufficient research of rate control for HEVC/H.265 and the complexities of algorithm and computation in HEVC rate control schemes are higher than that of previous standards. This obstacle has been hindering the adoption of HEVC/H.265 in its real-time streaming over mobile wireless networks. Therefore, it is necessary to develop low-complex and highly-efficient rate control schemes for H.265/HEVC to improve its network adaptability and enable its applications to various mobile wireless streaming scenarios [34]. Besides of rate control method, content aware segment length optimization (at GOP level) and Tile-based encoding schemes allow effective deployment of HEVC in MPEG DASH (Dynamic Adaptive Streaming over HTTP) [35]. This type of HEVC application is an emerging research area because of increasingly high traffic of high-quality live video over the Internet. However, there have been few related studies of this theme of HEVC for MPEG DASH applications.

The last theme of future research trends in HEVC could be predicted. There are numerous standards of video coding that are mutually incompatible (syntax and encoded data stream). Nevertheless, most of the supported coding Tools for standards are the same. The sole difference is the input parameters used for each of these tools in the specific standard. Hence, there may be a change of paradigm in development future codec named RVC. This new design paradigm allows implementing a set of common video coding functional blocks, then depending on the requirements of given standard, appropriate parameters are chosen for each block. Consequently, encoded bit-stream will consist of descriptive information of these blocks for decoding [30].

5. Conclusion

In this survey, the latest video coding standard HEVC/H.265 overview is given and its advancements, especially the detail developments of CABAC are discussed and summarized. The double coding efficiency of HEVC compared with its predecessor is the result of a series of modifications in most components and leads to a prospective integration of the standard into modern video applications. However, the significant increase in computation requirements, processing delay and consequently energy consumption hindered this progression. Literature review on hardware architecture and implementation strategies for highly efficient CABAC targeting high quality UHD resolution, real-time applications is provided. The challenges in the design and application of HEVC always exist as video applications are diverse due to human demands of progressive visual experience. Thus, the survey also addresses the challenges of utilizing HEVC in different video application areas and predicts several future research trends.

6. Acknowledgement

This work is partly supported by the Ministry of Science and Technology (MoST) of Vietnam under grant number 28/2018/ ĐTĐL.CN-CNC.

References

[1] Gary J. Sullivan et al, Overview of the High Efficiency Video Coding (HEVC) Standard, IEEE Transactions on Circuits and Systems for Video Technology 22 (12) (2012) pp. 1649-1668.

[2] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, T. Wiegand, High Efficiency Video Coding (HEVC) Text Specification Draft 9, document JCTVC-K1003, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Oct. 2012.

[3] F. Bossen, B. Bross, K. Suhring, D. Flynn. HEVC Complexity and Implementation Analysis, IEEE Transactions on Circuits and Systems for Video Technology 22 (12) 1685-1696.

[4] J- R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, T. Wiegand, Comparison of the Coding Efficiency of Video Coding Standard - including High Efficient Video Coding (HEVC), IEEE Transactions on Circuits and Systems for Video Technology 22 (12) (2012) 1669 -1684.

[5] P. Fröjdh, A. Norkin, R. Sjöberg, Next Generation Video Compression, Ericsson Review - The Communications Technology Journal 2013 (6) (2013), pp. 1-8.

[6] M. Wien, High Efficiency Video Coding - Coding Tools and Specifications, Signals and Communication Technology, Springer-Verlag Berlin Heidelberg, 2015.

[7] V.Sze et al, High Efficiency Video Coding (HEVC): Algorithms and Architectures, New York, Springer 2014.

[8] V. Sze Marpe, Entropy Coding in HEVC. Massachusetts Institute of Technology, 2014.

[9] V. Sze and M. Budagavi, A Comparision of CABAC Throughput for HEVC/H.265 vs. AVC/H.264, In Proceedings of the IEEE Workshop on Signal Processing Systems, Taipei, Taiwan, 16-18 October 2013, pp. 165-170.

[10] H. Shojania, S. Sudharsanan. A High Performance CABAC Encoder. Canadian Microelectronics Corporation (CMC) and Sun Microsystems, Inc.

[11] B. Peng et al, A Hardware CABAC Encoder for HEVC. In Proceedings of the IEEE International

Symposium on Circuits and Systems (ISCAS), 9-23 May 2013, pp. 1372-1375.

[12] B. Vizzotto, V. Mazui and S. Bampi, Area Efficient and High Throughput CABAC Encoder Architecture for HEVC, In Proceedings of the IEEE International Conference on Electronics, Circuits, and Systems (ICECS), Cairo, 2015, pp. 572-575.

[13] D. Kim et al, Hardware Implementation of HEVC CABAC Encoder, In Proceedings of the 2015 International SoC Design Conference (ISOCC), 2-5 November 2015, pp. 183-184.

[14] J. Zhou et al, A High-performance CABAC encoder architecture for HEVC and H.264/AVC, In Proceedings of the IEEE International Conference on Image Processing, Victoria, Australia, 15-18 September 2013, pp. 1568-1572.

[15] D-H. Pham, J. Moon, S. Lee, Hardware Implementation of HEVC CABAC Binarizer. Journal of IKEEE 18 (3) (2014) 356-361.

[16] Y. Cetin, A. Celebi, On the Hardware Implementation of Binarization for High Efficiency Video Coding, In Proceedings of Academicsera International Conference, Istanbul, Turkey, 23-24 October 2017.

[17] Quang-Linh Nguyen, Dinh-Lam Tran, Duy-Hieu Bui, Duc-Tho Mai, Xuan-Tu Tran, Efficient Binary Arithmetic Encoder for HEVC with Multiple Bypass Bin Processing, In Proceedings of the 7th International Conference on Integrated Circuits, Design, and Verification (ICDV), 5-6 October, Hanoi, Vietnam, 2017, pp. 82-87.

[18] V. Sze, M. Budagavi, High Throughput CABAC Entropy Coding in HEVC, IEEE Transactions on Circuits and Systems for Video Technology 22 (12) (2012) 1778-1791.

[19] D. Zhou et al. Ultra-high-throughput VLSI Architecture of H.265/HEVC CABAC Encoder for UHDTV Applications, IEEE Transactions on Circuits and Systems for Video Technology 25 (3) (2015) 497-507.

[20] C. Chen, K. Liu, S. Chen. High-Throughput Binary Arithmetic Encoder Architecture for CABAC in H.265/HEVC, In Proceedings of the 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT), Hangzhou, China, 2016, pp. 1416-1418.

[21] F-L-L. Ramos et al, High-Throughput Binary Arithmetic Encoder using Multiple-Bypass Bins Processing for HEVC CABAC, In Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 27-30 September 2018, pp. 1-5.

[22] H. Jo et al, Hardware Architecture of CABAC Binary Arithmetic Encoder for HEVC Encoder, Advanced Science and Technology Letters 141 (2016) 58-63.

[23] H. Jo, S. Park, K. Ryoo, High-Throughput Architecture of HEVC CABAC Binary Arithmetic Encoder, International Journal of Control and Automation 10 (5) (2017) 199-208.

[24] C-M. Alonso et al. Low-power HEVC Binarizer Architecture for the CABAC Block targeting UHD Video Processing, In Proceedings of the 30th Symposium on Integrated Circuits and Systems Design (SBCCI), August 28-September 01, 2017, pp. 30-35.

[25] F-L-L. Ramos et al. Low-Power Hardware Design for the HEVC Binary Arithmetic Encoder Targeting 8K Videos, In Proceedings of the 29th Symposium on Integrated Circuits and Systems Design (SBCCI), 29 August - 3 September 2016, pp. 1-6.

[26] F-L-L. Ramos et al, Novel Multiple Bypass Bins Scheme for Low-power UHD Video Processing HEVC Binary Arithmetic Encoder Architecture, In Proceedings of the 30th Symposium on Integrated Circuits and Systems Design (SBCCI), August 28-September 01, 2017, pp. 47-52.

[27] A. Adeyemi-Ejeye et al, Implementation of 4kUHD HEVC-Content Transmission 76 (17) (2017) 18099-18118.

[28] A. Ahmedin, A. Ghosh, D. Ghosal, A Survey of Multimedia Streaming in LTE Cellular Networks. https://arxiv.org/abs/1803.05080.

[29] K-E. Psannis. HEVC in Wireless Environments, Journal of Real-Time Image Processing 12 (2) (2016), pp. 443-454.

[30] D. Karwowski et al, 20 Years of Progress in Video Compression - from MPEG-1 to MPEG-H HEVC, General View on the Path of Video Coding Development, Springer International Publishing AG, 2017.

[31] J-H Hsieh, VLSI Design of an ML-Based Power-Efficient Motion Estimation Controller for Intelligent Mobile Systems, IEEE Transactions on VLSI Systems 26 (2) (2018) 262-271.

[32] T. Katayama, T. Song, T. Shimamoto, QP Adaptation Algorithm for Low Complexity HEVC based on a CNN-Generated Header Bits Map, In Proceedings of the IEEE 8th International Conference on Consumer Electronics - Berlin (ICCE-Berlin), 2018, pp. 1-5.

[33] M. Abdollahzadeh et al, Optimal HEVC Configuration for Wireless Video Communication

Under Energy Constraints, IEEE Access 6 (2018) 72479-72493.

[34] L. Chen et al, Framework and Challenges H.265/HEVC Rate Control in Realtime Transmission over 5G Mobile Networks, MOBIMEDIA 2017.

[35] C. Concolato et al, Adaptive Streaming of HEVC Tiled Videos using MPEG-DASH, IEEE Transactions on Circuits and Systems for Video Technology 28 (8) 2018, pp. 1981-1992.