electronics

MDPI

Article

An Efficient Hardware Implementation of Residual Data Binarization in HEVC CABAC Encoder

Dinh-Lam Tran [1], Xuan-Tu Tran [1,]*, Duy-Hieu Bui [1] and Cong-Kha Pham [2]

[1] VNU Key Laboratory for Smart Integrated Systems (SISLAB), University of Engineering and Technology, Vietnam National University, Hanoi 123106, Vietnam; lamtdvdt@gmail.com (D.-L.T.); hieubd@vnu.edu.vn (D.-H.B.)

[2] Department of Computer and Network Engineering, The University of Electro-Communications (UEC), Tokyo 182-8585, Japan; pham@ee.uec.ac.jp

* Correspondence: tutx@vnu.edu.vn

Received: 10 March 2020; Accepted: 21 April 2020; Published: 23 April 2020

Abstract: HEVC-standardized encoders employ the CABAC (context-based adaptive binary arithmetic coding) to achieve high compression ratios and video quality that supports modern real-time high-quality video services. Binarizer is one of three main blocks in a CABAC architecture, where binary symbols (bins) are generated to feed the binary arithmetic encoder (BAE). The residual video data occupied an average of 75% of the CABAC's work-load, thus its performance will significantly contribute to the overall performance of whole CABAC design. This paper proposes an efficient hardware implementation of a binarizer for CABAC that focuses on low area cost, low power consumption while still providing enough bins for high-throughput CABAC. On the average, the proposed design can process upto 3.5 residual syntax elements (SEs) per clock cycle at the maximum frequency of 500 MHz with an area cost of 9.45 Kgates (6.41 Kgates for the binarizer core) and power consumption of 0.239 mW (0.184 mW for the binarizer core) with NanGate 45 nm technology. It shows that our proposal achieved a high overhead-efficiency of 1.293 Mbins/Kgate/mW, much better than the other related high performance designs. In addition, our design also achieved a high power-efficiency of 8288 Mbins/mW; this is important factor for handheld applications.

Keywords: HEVC; CABAC; residual data binarization; hardware implementation

1. Introduction

HEVC (high efficiency video coding) standard has been designed to achieve multiple goals, including coding efficiency, ease of transport system integration, and data loss resilience. This 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% [1,2]. CABAC is the only entropy coding method applied in the final stage of the HEVC architecture, which contributes significantly to improving the coding efficiency of HEVC. Due to the high data dependency and sequential coding characteristic, it is addressed as the throughput bottleneck in the HEVC encoder [3]. The realization of HEVC in general and CABAC in particular in hardware to meet the criteria of preserving compression efficiency and high throughput is a big challenge for academics. Besides, power consumption and hardware resources are also other criteria to be considered in the development of HEVC to meet the demands for real-time, high quality, and battery-based video applications [4].

Over the past six years, there has been a significant effort from various research groups worldwide focusing on hardware solutions to improve the performance of HEVC codec in general and CABAC in particular [3]. Throughput, hardware cost, and power consumption are the main focusing design criteria of the state-of-the-art. Obviously, they are contrary and have to be traded off during designing for specific applications. To achieve these objectives, most of the previous works utilized several design strategies such as analyzing specific statistical features of video data [5–7], exploiting pipeline and parallelism, and multi-core architecture solutions [8,9]. The binarizer is the first component in CABAC architecture where syntax elements (SEs) from previous stages are analyzed and converted into bins that feed the binary arithmetic encoder (BAE). Thus, the performance of binarizer dramatically influences the whole CABAC [10]. Once the BAE, which is considered as the main bottle-neck in CABAC [11], received tremendous interests of academics and achieved a lot of performance enhancements, the binarizer could become one of the next throughput bottlenecks in CABAC. In fact, this issue has been focused recently to propose a higher performance binarizer.

Vizzotto et al. [12] proposed a heterogeneous binarization architecture for throughput improvement and area saving. Neji et al. [13] presented a binarizer architecture for CABAC addressing hardware complexity, regularity, and modularity. Alonso et al. [6] proposed a low-power binarizer architecture supporting ultra-high-definition (UHD) resolution while saving 20% of power consumption by analyzing statistical data for each binarization process in combination with a power-gating technique.

The workload of binarization in terms of SE types is also an interesting aspect that should be investigated to design high-performance binarizer in hardware. Obviously, as shown in Table 1, transform unit bins occupy a notable amount of data that the binarizer has to provide, 75% on the average and 94% in the worst-case [7]. Therefore, the performance of equivalent binarization processes of this SEs type greatly impacts on the CABAC binarizer. Transform unit bins are generated from residual SEs by several predefined binarization processes in the binarizer core. In the HEVC hierarchy, residual SEs are obtained from residual video data after undergoing prediction, transform and quantization steps. These data are in the form of transform block (TB) coefficients with different sizes from 4 × 4 up to 32 × 32 coefficients. Hence, a residual SE generation block needs to be placed between the transform-quantization stage and CABAC to convert these TB's coefficients into the residual SEs. Once the CABAC has been improved to support high throughput operation, its input data providers, especially residual SE generation, also need to be well-designed to meet its throughput requirement.

Table 1. Major bins contributors among HEVC data hierarchy [7].

	Common Test Condition				
Hierarchy Level	**AI**	**LD-P**	**LD-B**	**RA**	**Worst-Case**
Coding tree unit/coding unit bins	5.4%	15.8%	16.7%	11.7%	1.4%
Prediction unit bins	9.2%	20.6%	19.5%	18.8%	5.0%
Transform unit bins	85.4%	63.7%	63.8%	69.4%	94.0%

Note: The results are reported for each hierarchy level within the HEVC context: coding tree unit/coding unit, prediction unit, and transform unit. The common test criteria are used: all-intra (AI), low-delay P (LD-P), low-delay B (LD-B), and random access (RA).

In that view, this paper proposes a hardware design solution that implements a high-performance binarizer targeted to power and area saving while still providing high coding efficiency for high-quality video streams. Our main contributions are as follows:

- The procedure of residual SEs generation experiences multiple scans that require multiple accesses of TB memory. This operation causes high power consumption and increases processing delay. Thus, the performance of the residual SE generation would also influence the overall CABAC performance. For that reason, in this work we propose a residual SE generation algorithm and its hardware architecture which is able to reduce memory access load; therefore, it potentially leads to power saving.
- As one of the syntax elements in residual SEs set for each TB coefficient, the last significant coefficient position is represented by its X and Y coordinates. Theses coordinates are generated simultaneously and converted to bin strings by the same binarization method. Therefore, in this

paper, we propose an efficient design of combined SE binarization for these coordinates to save the area cost.

Following this introductory section, the remaining parts of this paper are organized as follows: Section 2 is a brief review on the CABAC principle and its general architecture, residual SE generation mechanism, the binarizer, and related binary processes for residual SEs. Section 3 presents in detail our proposed hardware architecture for residual SE binarization, and then hardware design strategies for area and power savings. The implementation results are presented and discussed in Section 4. Finally, the conclusion and future research are presented in Section 5.

2. Overview of Entropy Coding and Residual Binarization Algorithm in HEVC

2.1. General Architecture of CABAC

As mentioned above, CABAC is the sole entropy coding method used in HEVC and it is utilized at the last step of video encoding. It encodes the outputs of the previous stages, such as quantized transform coefficients, prediction modes, motion vectors, intra prediction direction, which are called syntax elements (SEs). CABAC architecture includes three main functional blocks: binarizer, context modeler, and BAE in addition to other modules such as buffers (FIFOs), multiplexers, and de-multiplexers as illustrated in Figure 1 [4].

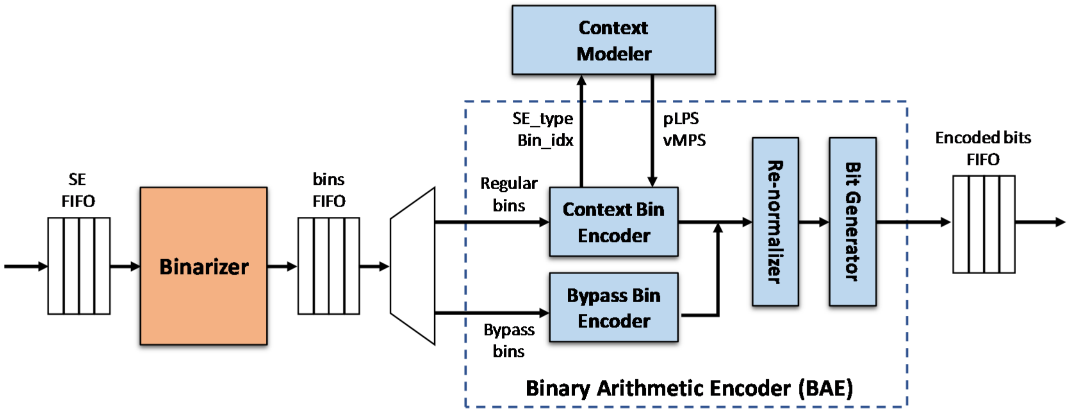


Figure 1. General hardware architecture of CABAC encoder.

In the first step, SEs are mapped into binary values at the binarizer. Depending on the type of SEs, an appropriate binarization method will be applied. HEVC standard defines several binary methods for binarization of all types of SEs. In the next step, context modeler provides the estimated probability of each bin. The probability of bins depends on the type of SEs it belongs to, the bin index within the SE (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 large finite state machine (FSM) for accurate context selection of each bin. In addition, the estimated probability of the selected context model is updated after each bin is encoded for the next bin context selection. Finally, the BAE compresses bins to bits based on the context model using the provided probability. As one of the main bottleneck functional blocks in CABAC [11], the BAE has attracted tremendous interests of researchers in the last few years and nowadays it achieved many performance enhancements. The binarizer becomes another throughput bottleneck block which influence the overall performance of the CABAC and needs to be improved.

2.2. Binarization in CABAC for HEVC Standard

The SEs from the other processes in HEVC have to be buffered at the input of CABAC encoder before feeding into the binarizer, where the process of mapping appropriate bins to each SE occurs.

In CABAC, the general hardware architecture of the binarizer can be characterized in Figure 2 [4]. Based on SE values and types, the controller will select an appropriate binarization process to produce bin string and bin length, accordingly. HEVC standard defines several basic binarization processes such as fixed length, truncated unary, truncated rice, and EGk (kth order exponential Golomb) for almost SEs. Some other SEs such as CALR (coefficient absolute level remaining) and QP_Delta (quantization parameter offset value) utilize two or more combinations (prefix and suffix) of these basic binarization processes [14,15]. There are also simplified custom binarization formats that are mainly based on LUT (look-up table), for other SEs like inter-pred mode, intra-pred mode, and part mode.

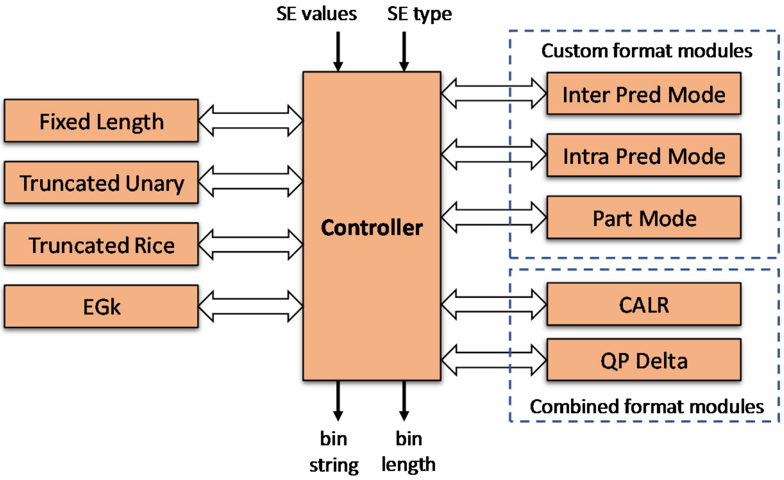


Figure 2. General hardware architecture of a binarizer [4].

2.3. Residual Syntax Generation and Binarization

HEVC supports different coding unit sizes in general and different residual TB sizes, ranging from 4 × 4 up to 32 × 32 pixels. These TBs are converted into equivalent residual transformed coefficients by transformation and quantization steps. At this stage, instead of zigzag scan pattern used in H.264/AVC [16], HEVC utilizes the diagonal scan pattern, that traverses all TBs to convert these 2D blocks of residual transformed coefficients into the 1D arrays. The diagonal scan pattern starts from the bottom-right of all types of TBs and scans up to the top-left of that TB.

The diagonal scan is applied to divide large TBs into un-overlapped 4 × 4 sub-blocks of coefficients. Then the scan occurs within each 4 × 4 block to form a 1-D array of 16 consecutive coefficients, which is called a coefficient group. The process of diagonal scanning is illustrated in Figure 3 [17].

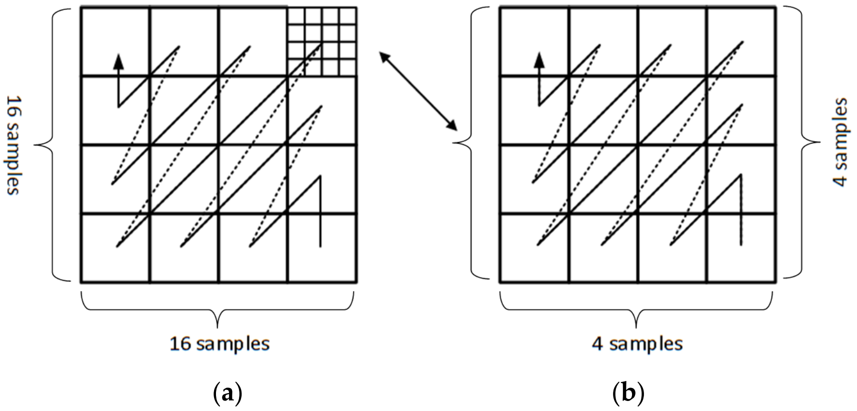


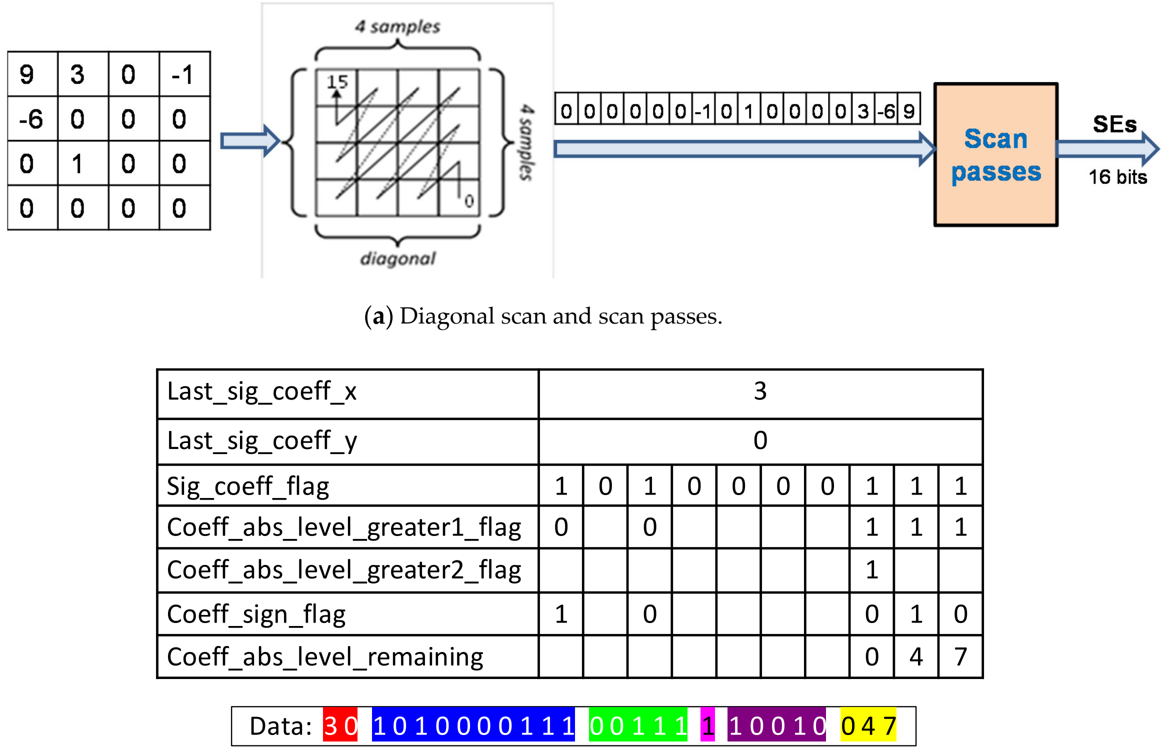
Figure 3. Diagonal scanning: (**a**) in large transform block, and (**b**) within 4 × 4 transform block.

At this coefficient group, different scan passes are applied to generate residual SEs. The last significant coefficient SE is determined first and this is the entry point of five other scan passes to form the remaining residual SEs. Table 2 summarizes these SEs to represent residual transformed data of every 4 × 4 TB (i.e., coefficient group).

Table 2. Syntax elements of 4 × 4 residual transform data.

Syntax Element	Descriptions
Last_significant_coeff	The first non-zero coefficient in scanning order within coefficient group.
Significant_coeff_flag	Significance of a coefficient (zero/non-zero).
Coeff_abs_level_greater1_flag	Flags indicating whether the absolute value of a coefficient level is greater than 1.
Coeff_abs_level_greater2_flag	Flag indicating whether the absolute value of a coefficient level is greater than 2.
Coeff_sign_flag	Sign of a significant coefficient (0: positive; 1: negative).
Coeff_abs_level_remaining	Remaining value for the absolute value of a coefficient level.

The process of scanning and forming the above residual SEs can be illustrated in Figure 4 [7]. Figure 4a describes how the diagonal scan pattern is applied on a specific 4 × 4 TB to generate a coefficient group, which will be undergone scan passes to extract residual SEs. Figure 4b shows detailed extracted SEs and the ordered data sequence of those SEs at the input of the binarization stage of CABAC.



9	3	0	-1
-6	0	0	0
0	1	0	0
0	0	0	0

(**a**) Diagonal scan and scan passes.

Last_sig_coeff_x	3									
Last_sig_coeff_y	0									
Sig_coeff_flag	1	0	1	0	0	0	0	1	1	1
Coeff_abs_level_greater1_flag	0		0					1	1	1
Coeff_abs_level_greater2_flag								1		
Coeff_sign_flag	1		0					0	1	0
Coeff_abs_level_remaining								0	4	7

Data: 3 0 1 0 1 0 0 0 0 1 1 1 0 0 1 1 1 1 1 0 0 1 0 0 4 7

(**b**) Generated SEs and output order.

Figure 4. Process of residual SEs generation.

2.4. State-of-the-Art

Once the most throughput bottleneck in CABAC (i.e., BAE) [4,11], has been improved significantly throughout a large number of researches to meet the demand of UHD video sequences, the binarizer then is considered to become the potential throughput bottleneck. Thus, it becomes the candidate for numerous researches that focus on designing efficient binarizer hardware architectures while maintaining the throughput requirement of the whole CABAC encoder. Vizzotto et al. [12] proposed an area efficient and high throughput CABAC encoder, where almost CABAC components are optimized to support throughput improvement. In addition, a heterogeneous binarization core is proposed to support UHD applications while reducing area cost of up to 10 Kgates in comparison with traditional parallel binarization architecture. A completed binarizer architecture is proposed by Neji et al. [13] that supports regularity and modularity features to be able to be used in different standards (HEVC/H.264). Alonso et al. [6] proposed an HEVC binarizer architecture that uses parallel multi-core design strategy to meet the throughput requirement of UHD video sequences. Whereas, the statistical analysis of video data is applied to implement the operand isolation algorithm into the architecture of each binarizer core to save about 20 ÷ 22% of power consumption. Some other works focus on the process of residual SE generation, which is the largest workload of binarizer. Saggiorato et al. [10] proposed a novel efficient multi-core architecture of SE generation, resulting in up to 4 pipelined SE processing cores to avoid binarizer input starved issue. CABAC input starvation is also raised by Ramos et al. in [7] and a multi-core SE generation architecture is also proposed. In addition, they also proposed a power-saving design strategy throughout analyzing the occurrence frequency of each specific residual SE. Their proposal archives around 30% of power gain compared to the original design. However, most of the above works require multiple accesses to TB memory and high hardware cost. Therefore, we propose a hardware design solution that implements a high-performance binarizer targeted to power saving and area saving while still ensures coding efficiency and meets the requirements of high-quality video streams. This proposal includes: (i) a residual SE generation algorithm and its hardware architecture which is able to reduce memory access load which potentially leads to power saving; and (ii) an efficient design of combined SE binarization to save the hardware area.

3. Proposed Hardware Architecture and its Implementation for Residual Binarization

3.1. Overall Hardware Architecture with an Efficient Scanning Algorithm

The overall block diagram of the binarization module for residual data processing is composed of SE generation and binarizer core, as shown in Figure 5. The SE generation process involves scanning passes to generate residual SEs as previously described in Figure 4. Then, they are appropriately delivered to and invoked by an equivalent sub-binarization process to generate bin string of each SE. In addition, the binarizer core will output the final bin string of these SEs.

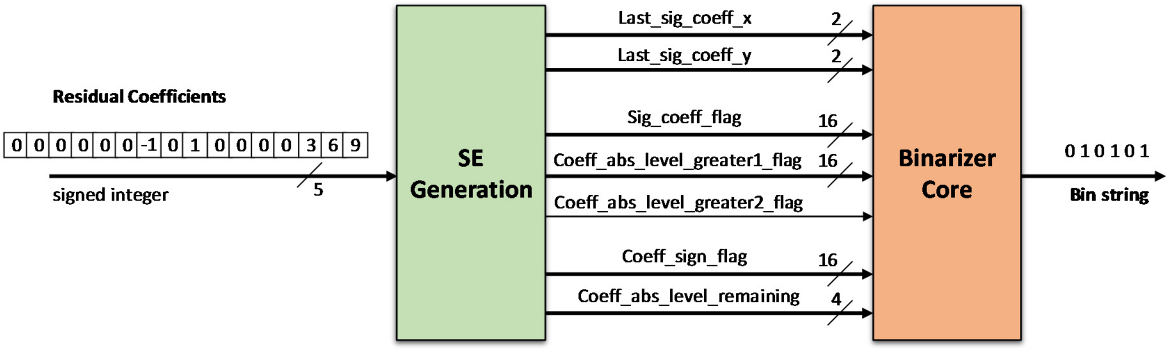


Figure 5. Overall block diagram of residual SE binarization module.

In the SE generation stage, after the position of the last significant coefficient (the first non-zero coefficient) is determined, each coefficient group is up-right diagonal scanned throughout five scan passes to determine the remaining SEs as listed in Table 2. Except for the last significant coefficient and CALR SEs, the remaining SEs are the flags with one-bit values. Moreover, these SEs are converted into bins by the same binarization process using the fixed-length binarization. Therefore, an efficient architecture of the syntax element generation is proposed as shown in Figure 6.

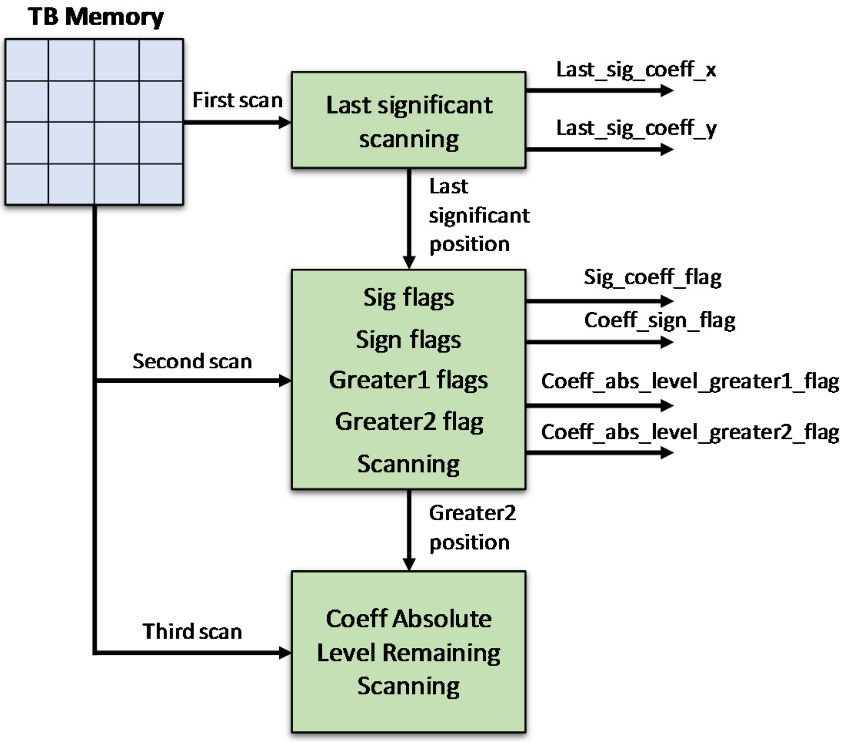


Figure 6. Scanning and SE generation architecture.

As described in [16], each coefficient group requires 4 scan passes to generate the flagged-SEs, which consequently need four accesses to transform block memory. This may lead to processing latency and power consumption issues for the whole binarization design. In our proposed method, instead of scanning the coefficient group four times, it is scanned all at once for determining all the four flagged-SEs in a single datapath. This will reduce the number of memory accesses when compared with the traditional methods.

Figure 7 shows the proposed hardware implementation of the architecture in Figure 6 for residual SE generation. In this architecture, the last significant position of the coefficient group is determined firstly for Last_sig_coeff_x, Last_sig_coeff_y SE generations and becomes the entry point for the next scans. From this position, each coefficient is evaluated to generate significant, sign, greater-than-one, and greater-than-two flags that are complied with the HEVC standard. For significant flagged-SEs, each coefficient is evaluated whether equal to zero to add a '1' or '0' flag to its flag vector. For each non-zero coefficient, a comparison with zero to determine whether it is positive or negative one to add a '0' or '1' flag to sign flag vector. This sign flag is also used to calculate the absolute value of the coefficient, which will be used to determine the greater-than-one and greater-than-two flagged-SEs as well.

3.2. Combined SE Binarization Hardware for Low Area Cost

Generated SEs for 4 × 4 TB or 4 × 4 sub-block of residual coefficients from SE generation module are sent to the binarization core. Based on each SE type, it will invoke an appropriate binarization process to generate bin string. All flagged-SEs are encoded with Fixed Length process and CALR SEs are encoded by CALR process. Whereas X and Y coordinates of the last significant coefficient position are undergone truncated rice binary processes. Normally, they are simultaneously binarized by two separate truncated rice modules. Hardware implementation of this module for X and Y coordinates is shown in Figure 8.

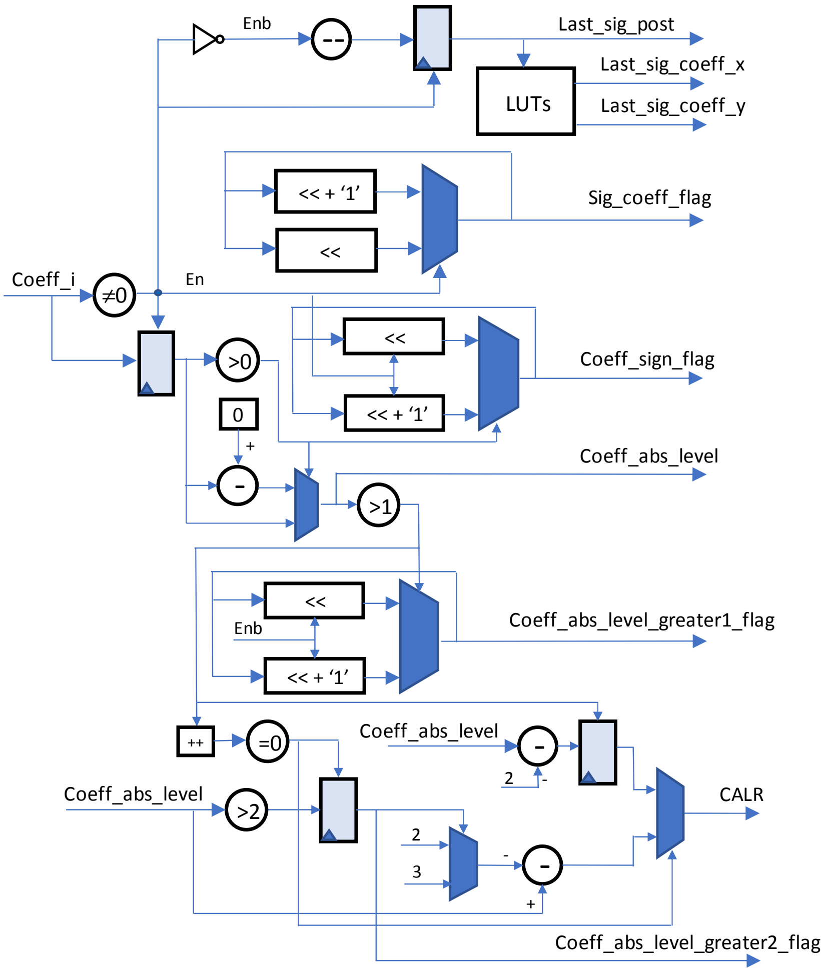


Figure 7. Proposed hardware implementation of residual SE generation.

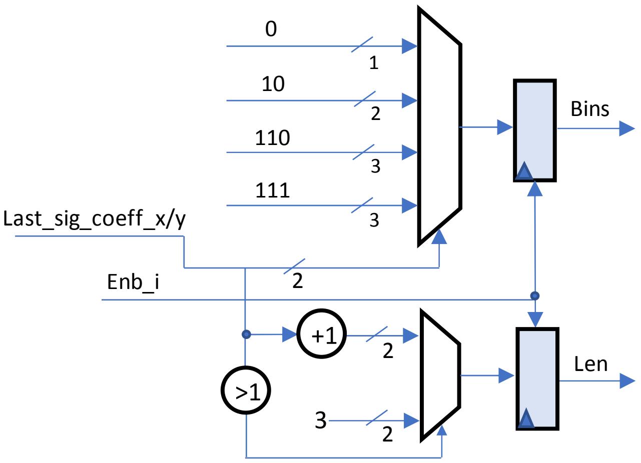


Figure 8. Truncated rice binarization hardware architecture for X or Y coordinates.

In the hardware implementation, if these two SEs are combined to consecutively process on one datapath by invoking only one truncated rice module, there will be a reduction of one truncated rice module in comparison with conventional architectures. Therefore, we proposed efficient hardware architecture of the binarization core for residual SEs as illustrated in Figure 9.

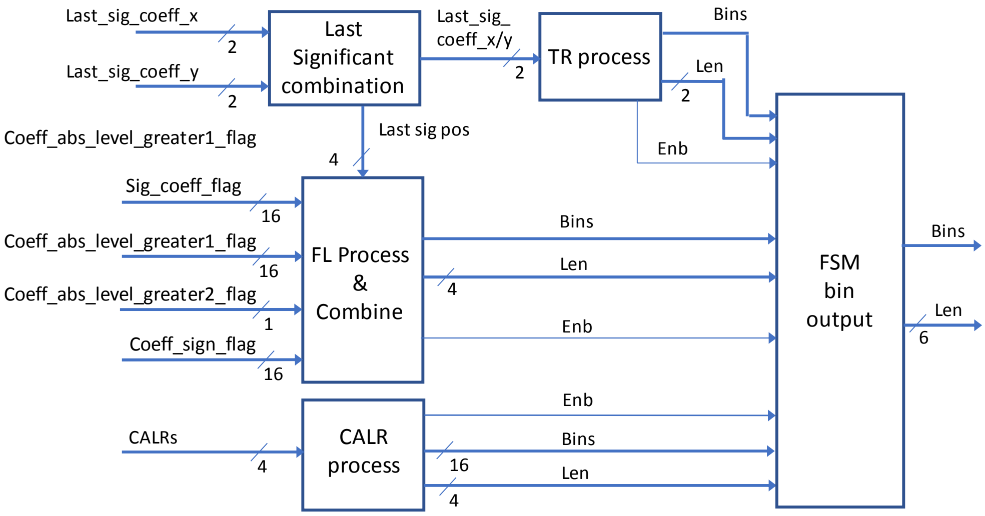


Figure 9. Hardware architecture of residual binarizer.

An FSM bin output module will be included to manipulate the output order of bin string of all residual SEs as depicted in Figure 9. In the binarizer core, we proposed a combined Last X and Y coordinates truncated rice binarization sub-module to process consecutive coordinates at the same datapath. The proposed hardware implementation for this sub-module is shown in Figure 10. The enabled input signal, which comes along with Last_sig_coeff_x and Last_sig_coeff_y, will be used to select input coordinates properly for the truncated rice core and control the outputs as well. Obviously, this proposed hardware architecture consumes less hardware resource than the architecture of two truncated rice cores running separately.

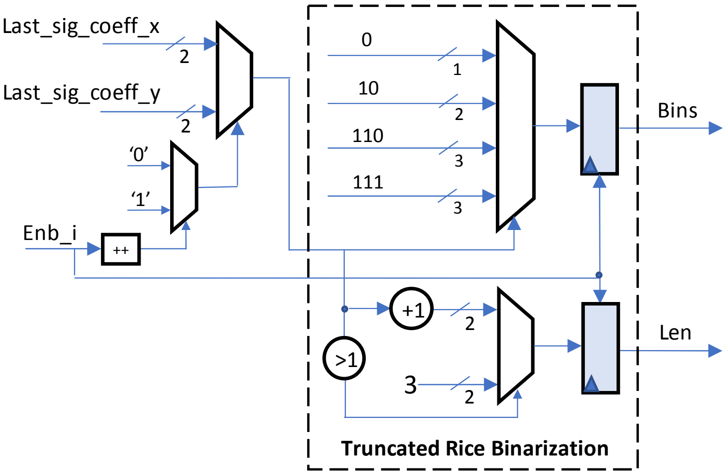


Figure 10. Proposed combined last X-Y significant binarization architecture.

4. Experimental Results and Comparisons

The proposed hardware architecture is modeled in VHDL and tested with different cases of 4 × 4 TBs and 4 × 4 sub-blocks of residual data. The testing results show that our proposed architecture can process 3.5 SEs/cycle (3.05 bins/cycle) on average at the maximum operating frequency of 500 MHz. The proposed design is capable of generating bins for a binary arithmetic encoder of UHD video sequences. Then, its hardware architecture is synthesized by a Synopsys design compiler using NanGate 45 nm technology. Our architecture has a total gate count of 9.45 Kgates, a power consumption of 0.239 mW at the operating frequency of 500 MHz.

Table 3 shows the comparisons with the prior state-of-the-art which focuses on the performance of the binarizer and SE generation for residual data. The table shows that the work of Alonso [6] achieved highest throughput, then the works in [7,8] and our proposal. The work of Alonso [6] achieved a thoughput of 8.34 bins/cycle at the operating frequency of 834 MHz (i.e., achieved a throughput of 6956 Mbins/s) as it is a parallel four-core binarizer architecture designed to process 8K-UHD video sequences. This achievement comes at the cost of large hardware area and power consumption. In contrast, our binarizer core targeting low hardware cost with acceptable performance can process 3.05 bins/cycle on the average at the operating frequency of 500 MHz (i.e., 1525 Mbins/s) that can generate bins for CABAC to support UHD video sequences. At this operating frequency, our binarizer core occupies 6.41 Kgates, which is nearly a half of the area cost in [6] with the power consumption is only one-tenth of that in [6].

Table 3. Comparisons with related works.

	Kim 2015 [18]	**Alonso 2017** [6]	**Peng 2013** [5]	**Vizzotto 2015** [12]	**Zhou 2015** [8]	**Pham 2014** [14]	**Ramos 2019** [7]	**Our Work**
								SE Gen + Bin Core
Standard	HEVC	HEVC	HEVC	HEVC	HEVC	HEVC	HEVC	**HEVC**
Technology process (nm)	180	65	130	130	90	45	65	**45**
Clock frequency (MHz)	158	**834**	357	380	**420**	200	**668**	**500**
Gate count (Kgates)	**3.41**	11.85	48.94	31.18	64.1	**1.678**	**3.67**	**9.45 (6.41 binarizer core only)**
Throughput (bins/cycle)	1 bin/cycle	**8.34 bins/cycle** (4 SEs/cycle)	1.18 bins/cycle	2.37 bins/cycle (6 SEs/cycle)	**4.36 bins/cycle** (2 ÷ 4 SEs/cycle)	1 bins/cycle	**4.5 bins/cylcle**	**3.05 bins/cycle (3.5 SEs/cycle)**
Throughput (Mbins/s)	158	**6956**	421	901	**1835**	200	**2672**	**1525**
Power consumption (mW)	-	**1.87**	-	-	-	**0.05325**	11.52	**0.239 (0.184 binarizer core)**
Resolution	1920 × 1080	8K UHD	2560 × 1600	UHD	8K UHD	1920 × 1080	2560 × 1600	**UHD**
Area-Efficiency (Mbins/Kgate)	0.046	**0.587**	0.009	0.029	0.029	0.119	**0.728**	**0.238**
Overhead-Efficiency (Mbin/Kgate/mW)	-	0.314	-	-	-	**2.238**	0.063	**1.293**
Power-Efficiency (Mbins/mW)	-	**3719.551**	-	-	-	**3755.869**	231.944	**8288.043**
Notes	Binarizer	Binarizer	CABAC	CABAC	CABAC	Binarizer	Residual SE Generation	**Residual SE Generation & Binarizer**

Our proposed method in residual SE generation design is for power saving purposes via reducing memory access load. Its throughput of 3.5 SEs/cycle at 500 MHz is comparable with the results of [6,8,12]. The efficiency of our proposal in terms of throughput is a half of [7] as they proposed four parallel residual SE generation cores working at the operating frequency of 668 MHz. However, [7] shows the significantly higher power consumption compared to ours. Most of this power consumption (over 90%) is the dynamic power consumption due to four cores accessing memory at the same time. In addition, our method uses less memory access time to determine SEs in each 4 × 4 sub-block scanning than the work in [7].

If we consider the area-efficiency of the binarizer (calculated by the throughput-area cost ratio), the work in [7] is the most efficient design, then the work in [6] and our proposal; they achieve 0.728, 0.587, and 0.238 Mbins/Kgate, respectively. However, if we consider the overhead-efficiency, which is calculated by the ratio between the achieved throughput and the total overhead (both area cost and power consumption), the work in [14] is the most efficient design (2.238 Mbins/Kgate/mW), then our proposed design (1.293 Mbins/Kgate/mW) and the work in [6] (0.314 Mbins/Kgate/mW). However, the work in [14] is just a low-performance low-cost binarization core, which is not able to support UHD applications. Its throughput is only 200 Mbins/s while our design's throughput is 1525 Mbins/s (7.6 times higher). Therefore, between the high-throughput designs [6–8,12], our proposal is the best (having a throughput-overhead efficiency of 1.293 Mbins/KGate/mW), 4× better than the work in [6] and 20× better than the work in [7]. In addition, if consider the power-efficiency (calculated by the throughput-power consumption ratio), our proposal is the most efficient design even compared with the work in [14]. Our design achieved a very high power-efficiency of 8288 Mbins/mW while the works in [6,14] have a similar power-efficiency (3720 and 3756 Mbins/mW)—about a half of our design's power-efficiency.

5. Conclusions and Future Work

To improve the overall performance of HEVC CABAC encoder, beside the binary arithmetic encoding, the design of binarization and syntax element generation should be carefully investigated, particularly for residual data encoding process. In this paper, we have focused on analyzing the CABAC binarizer workload, where residual SEs occupy a significant portion, evaluating potential strategies to effectively process and implement the binarizer hardware for these SEs. Then, we have proposed an efficient hardware implementation for syntax element generation and binarization for residual data to meet the throughput demand of CABAC. One side, the proposed architecture includes a scanning strategy to reduce the memory access times; therefore, saving power consumption for SE generation. On the other side, the hardware cost of our design has been significantly reduced thanks to the combined SE hardware architecture of last significant SEs. The complete hardware architecture of binarization and syntax element generation has been modeled in VHDL and synthesized by synopsys design compiler with NanGate 45 nm technology. It achieved a throughput of 3.5 SEs/cycle at the maximum operating frequency of 500 MHz with an area cost of 9.45 Kgates (6.41 Kgates for the binarizer core) and power consumption of 0.239 mW (0.184 mW for the binarizer core). Compared to other related works, our proposal achieved outstanding high efficiency, 1.293 Mbins/Kgates/mW in terms of total overhead-efficiency and 8288 Mbins/mW in terms of power-efficiency.

Author Contributions: Conceptualization, X.-T.T.; Data curation, D.-L.T.; Formal analysis, D.-L.T.; Investigation, D.-L.T. and X.-T.T.; Methodology, X.-T.T.; Project administration, X.-T.T.; Resources, D.-L.T. and D.-H.B.; Software, D.-L.T. and D.-H.B.; Supervision, X.-T.T.; Validation, D.-L.T. and X.-T.T.; Visualization, D.-L.T.; Writing—original draft, D.-L.T. and X.-T.T.; Writing—review & editing, X.-T.T., D.-H.B. and C.-K.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partly supported by Vietnam National University, Hanoi (VNU) under grant number QG.18.38 (LoPoSoC).

Acknowledgments: The authors would like to thanks the editors and reviewers for their helpful comments to improve the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bossen, F.; Bross, B.; Suhring, K.; Flynn, D. HEVC Complexity and Implementation Analysis. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1685–1696. [CrossRef]
2. Ohm, J.-R.; Sullivan, G.J.; Schwarz, H.; Tan, T.K.; Wiegand, T. Comparison of the Coding Efficicency of Video Coding Standard—Including High Efficient Video Coding (HEVC). *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1669–1684. [CrossRef]

3. Sze, V.; Budagavi, M. A Comparison of CABAC Throughput for HEVC/H.265 vs. AVC/H.264. In Proceedings of the IEEE Workshop on Signal Processing Systems, Taipei City, Taiwan, 16–18 October 2013.
4. Tran, D.-L.; Pham, V.-H.; Nguyen, K.H.; Tran, X.-T. A Survey of High-Efficient CABAC Hardware Implementations in HEVC Standard. *VNU J. Comput. Sci. Commun. Eng.* **2019**, *35*, 1–21.
5. Peng, B.; Ding, D.; Zhu, X.; Yu, L. A Hardware CABAC Encoder for HEVC. In Proceedings of the IEEE International Symposium on Circuits and Systems, Beijing, China, 19–23 May 2013.
6. Alonso, C.-M.; Ramos, F.-L.-L.; Zatt, B.; Porto, M.; Bampi, S. 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, Fortaleza, Brazil, 28 August–1 September 2017.
7. Ramos, F.-L.-L.; Saggiorato, A.-V.-P.; Zatt, B.; Porto, M.; Bampi, S. Residual Syntax Elements Analysis and Design Targeting High-Throughput HEVC CABAC. *IEEE Trans. Circuits Syst.* **2019**, *67*, 475–488. [CrossRef]
8. Zhou, D.; Zhou, J.; Fei, W.; Goto, S. Ultra-high-throughput VLSI Architecture of H.265/HEVC CABAC Encoder for UHDTV Applications. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 497–507. [CrossRef]
9. Ramos, F.-L.-L.; Zatt, B.; Porto, M.; Bampi, S. 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, Florence, Italy, 27–30 May 2018.
10. Saggiorato, A.-V.-P.; Ramos, F.-L.-L.; Zatt, B.; Porto, M.; Bampi, S. HEVC Residual Syntax Elements Generation Architecture for High-Throughput CABAC Design. In Proceedings of the 25th IEEE International Conference on Electronics, Circuits and Systems, Bordeaux, France, 9–12 December 2018.
11. Nguyen, Q.-L.; Tran, D.-L.; Bui, D.-H.; Mai, D.-T.; Tran, X.-T. Efficient Binary Arithmetic Encoder for HEVC with Multiple Bypass Bin Processing. In Proceedings of the 7th International Conference on Integrated Circuits, Design, and Verification, Hanoi, Vietnam, 5–6 October 2017.
12. Vizzotto, B.; Mazui, V.; Bampi, S. Area Efficient and High Throughput CABAC Encoder Architecture for HEVC. In Proceedings of the IEEE International Conference on Electronics, Circuits, and Systems, Cairo, Egypt, 6–9 December 2015.
13. Neji, N.; Jridi, M.; Alfalou, A.; Masmoudi, N. FPGA Implementation of Improved Binarizer Design for Context-Based Adaptive Binary Arithmetic Coder. In Proceedings of the International Image Processing, Applications and Systems, Hammamet, Tunisia, 5–7 November 2016.
14. Pham, D.-H.; Moon, J.; Le, S. Hardware Implementation of HEVC CABAC Binarizer. *J. IKEEE* **2014**, *18*, 356–361. [CrossRef]
15. Cetin, Y.; Celebi, A. On the Hardware Implementation of Binarization for High Efficiency Video Coding. In Proceedings of the Academicsera International Conference, Istanbul, Turkey, 23–24 October 2017.
16. Sole, J.; Joshi, R.; Nguyen, N.; Ji, T.; Karczewicz, M.; Clare, G.; Henry, F.; Duenas, A. Transform Coefficient Coding in HEVC. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1765–1777. [CrossRef]
17. Nguyen, T.; Helle, P.; Winken, M.; Bross, B.; Marpe, D. Transform Coding Techniques in HEVC. *IEEE J. Sel. Top. Signal Process.* **2013**, *7*, 978–989. [CrossRef]
18. Kim, D.; Moon, J.; Lee, S. Hardware Implementation of HEVC CABAC Encoder. In Proceedings of the International SoC Design Conference, Gyungju, Korea, 2–5 November 2015.