Design and Implementation of Parallel Bypass Bin Processing for CABAC Encoder

Nagaraju MAMIDI, Santosh Kumar GUPTA, Vijaya BHADAURIA

Department of Electronics & Communication Engineering, Motilal Nehru National Institute of Technology, MNNIT Allahabad Campus, Teliarganj, 211004 Prayagraj, Uttar Pradesh, India

rel1651@mnnit.ac.in, skg@mnnit.ac.in, vijaya@mnnit.ac.in

DOI: 10.15598/aeee.v19i3.4010

Article history: Received Nov 18, 2020; Revised May 31, 2021; Accepted Jun 10, 2021; Published Sep 30, 2021.

Abstract. *The ever-increasing demand for high-quality digital video requires efficient compression techniques and fast video codecs. It necessitates increased complexity of the video codec algorithms. So, there is a need for hardware accelerators to implement such complex algorithms. The latest video compression algorithms such as High-Efficiency Video Coding (HEVC) and Versatile Video Coding (VVC) have been adopted Context-based Adaptive Binary Arithmetic Coding (CABAC) as the entropy coding method. The CABAC has two main data processing paths: regular and bypass bin path, which can achieve good compression when used with Syntax Elements (SEs) statistics. However, it is highly intrinsic data dependence and has sequential coding characteristics. Thus, it is challenging to parallelize. In this work, a 6-core bypass bin path having high-throughput and low hardware area has been proposed. It is a parallel architecture capable of processing up to 6 bypass bins per clock cycle to improve throughput. Further, the resource-sharing techniques within the binarization and a common controller block have reduced the hardware area. The proposed architecture has been simulated, synthesized, and prototyped on 28 nm Artix 7 Field Programmable Gate Array (FPGA). The implementation of Application Specific Integrated Circuit (ASIC) has been done using 65 nm CMOS technology. The proposed design achieved a throughput of 1.26 Gbin·s^{-1} at 210 MHz operating frequency with a low hardware area compared to existing architectures. This architecture also supports multi-standard (HEVC/VVC) encoders for Ultra High Definition (UHD) applications.*

Keywords

ASIC, bypass bin, CABAC, FPGA, HEVC, VVC.

1. Introduction

In the recent years, storing and transmitting a considerable amount of video data has become one of the most significant video processing challenges. The video compression technique is one of the solution used to reduce the video data size. However, the next-generation video compression techniques are expected to support UHD (4K and above) and 360° video [1]. These require high-throughput and area-efficient compression techniques to store and transmit video data.

International Telecommunication Union-Telecommunication (ITU-T)/ Video Coding Experts Group (VCEG), International Standardization Organization/ International Electrotechnical Commission (ISO/IEC), and Moving Picture Experts Group (MPEG) are the international organizations involved in developing the video coding standards. The H.264 or MPEG-4 AVC (MPEG-4 part 10, Advanced Video Coding) video standard was jointly developed by ITU-T and ISO/IEC in 2003 [2]. This video coding standard has been among the widely used video formats in the last decade.

Recently, the UHD has gained popularity because of its enhanced video quality. However, supporting UHD videos requires higher coding efficiency than H.264 or MPEG-4 AVC [3]. The HEVC [3] and VVC [4] standards supporting UHD have been released jointly by ITU-T and ISO/IEC in 2013 and 2020, respectively. The HEVC offers an efficient compression level than its predecessor H.264. It is predominantly suited to higher-resolution video streams where bandwidth efficiency is about 50 % less than H.264 [3]. The VVC is the next generation and successor of the HEVC video standard [4].

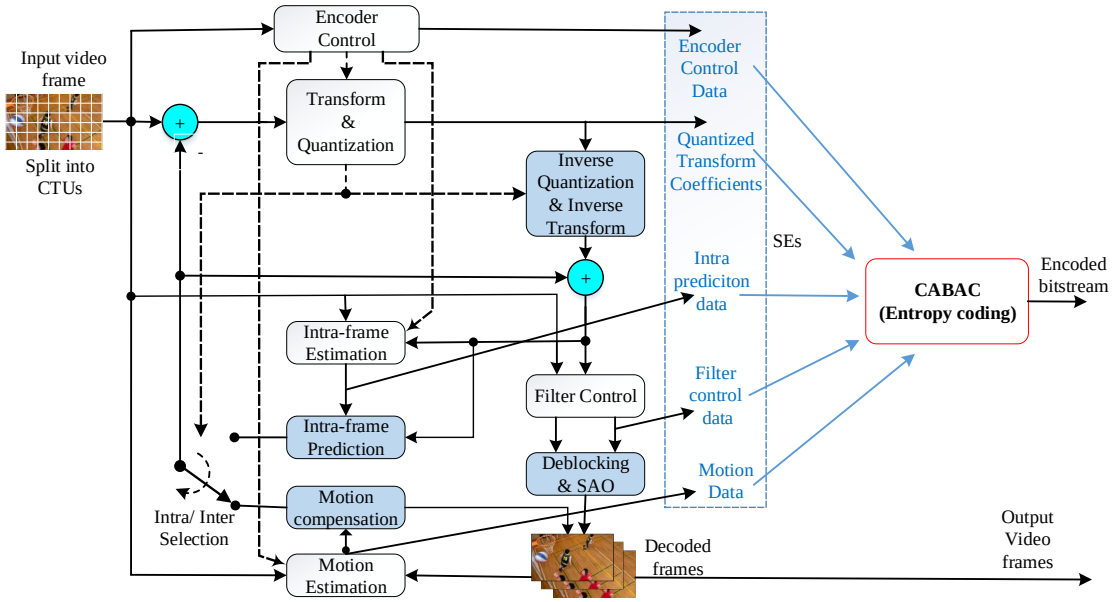

Fig. 1: Block diagram of a typical HEVC video encoder [3].

Tab. 1: Video standards and corresponding entropy coding.

Video standard	**MPEG-2 [2]**	**H.264/ AVC [2]**	**H.265/ HEVC [3]**	**H.266/ VVC [4]**
Year of release	1994	2003	2013	2020
Entropy coding	VLC*	CAVLC/ **CABAC**	**CABAC**	**CABAC**
Primary support	SD**	2K	4K	8K

*VLC-Variable Length Coding, **SD-Standard Definition

These standards are designed to achieve various goals, including coding efficiency, high-throughput, high processing speed, fewer hardware resources, and low power consumption to meet higher resolution demands, high frame rates, and low power applications.

In HEVC, video encoding is performed using a basic block called the Coding Tree Units (CTUs) made up of distinct sizes (8×8 to 64×64) and are obtained by dividing each frame [5]. These CTUs are processed through several internal coding blocks, such as Transform, Quantization, Intra-frame estimation and prediction, Motion compensation, Motion estimation, Deblocking, Sample Adaptive Offset (SAO) filter, and CABAC, as shown in Fig. 1.

The final stage of the encoder is CABAC, which does entropy coding. Entropy coding is a lossless compression technique that removes statistical redundancy and increases the overall encoder ratio of compression efficiency. It is the last step of encoding and the first step of the decoding process. CABAC is one of the entropy method that resembles Context-based Adaptive Variable Length Coding (CAVLC). Still, it provides a considerably higher compression efficiency than others [6]. Table 1 shows the evolution of different video standards and the corresponding entropy coding.

Among the standards mentioned in Table 1, CABAC has emerged as the efficient entropy coding method for the next-generation video standards also. However, due to several inherent feedback loops in its architecture, its parallelizing and pipelining become complicated and it is well known as a throughput bottleneck in the video encoder implementation [7]. Besides, it leads to high computation and hardware complexity, especially for UHD applications. Since the inception of the CABAC algorithm, several researchers have focused on hardware architectures that act as trade-offs between coding efficiency, high-throughput, low hardware area, and low power consumption. An efficient hardware architecture for the CABAC-bypass bin path combining parallel processing and resource-sharing techniques has been proposed in this work. It offers a high-throughput and reduced hardware area using efficient memory management without any degradation in functionality. The main contributions of this work are as follows:

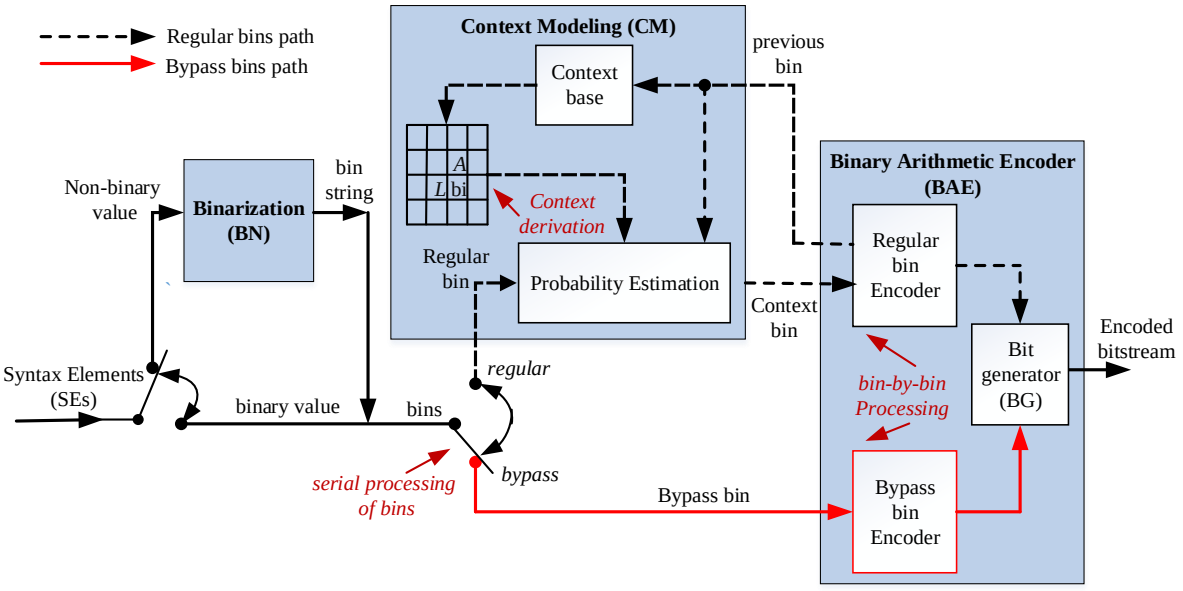

Fig. 2: CABAC encoder general block diagram [6] and [7].

- A highly parallelized and resource-sharing architecture for the CABAC-bypass bin path has been proposed. The focuses are given to high-throughput and area-efficient hardware. The proposed architecture ensures efficient memory management by the use of storage buffers in the data path.
- A Bypass Bin Splitting (BBS) scheme that allows parallelism of a grouped bypass bins in the path has been used, which effectively helped to achieve higher Bins Per Clock Cycle (BPCC) processing.
- It is modeled in Verilog HDL, synthesized & verified on FPGA, and implemented on the ASIC platform. It has also been tested with the standard UHD video sequences.

The rest of the paper is organized as follows. Sec. 2. introduces the CABAC process and the literature survey. Section 3. deals with the proposed architecture and its sub-blocks. Section 4. presents the experimental test setup and implementation. The results and discussions are presented in Sec. 5. Finally, Sec. 6. concludes the paper.

2. Basics of CABAC Process and Literature Survey

It is to be noted that CABAC provides high coding efficiency, but its serial process of bins and critical bin-by-bin data dependencies cause it to be a throughput bottleneck for the video encoder [7]. Conventionally, the CABAC consists of three main blocks, as shown in Fig. 2: (i) Binarization (BN), (ii) Context Modeling (CM), and (iii) Binary Arithmetic Encoder (BAE).

The CABAC input data comes from the previous encoding steps as Syntax Elements (SEs) (Fig. 1), namely: Encoder control, Quantized transform coefficients, Intra-frame prediction, and Motion data. First, the data of non-binary SEs are converted into bins (binary symbols) through the BN block. Then, the probability estimation of each bin is determined based on the surrounding information of a particular bin and the previously coded bins through the CM block. Finally, the BAE block converts bins into an encoded bitstream using the recursive interval division principle [6].

The BN is the first step of the CABAC process. The general block diagram of a BN block is shown in Fig. 3. In this stage, a bin string is generated for each non-binary SE input. The arithmetic coder in the CABAC engine can only encode binary symbol values. Thus, the BN process needs to convert the non-binary SE into a binary. The BN process receives the SEs with SE_value and SE_type at the input. It gets the data of all SEs and translates them to a new symbolism (bin string) according to pre-defined methods of the video standard. Based on SE_type, the controller selects an appropriate BN method to convert the SE_value into bin string and produces bin length accordingly.

The BN process operates in one of the three formats: (i) Single, (ii) Combined, and (iii) Custom. In a single format, the BN is performed using one of the primary methods, namely, Fixed Length (FL), Truncated Unary (TU), Truncated Rice (TR) and Exp-Golomb kth order (EGk).

Tab. 2: The primary binarization methods [7].

SE value	Fixed-Length (FL) $cMax = 7$	Truncated Unary (TU) $cMax = 7$	Truncated Rice (TR) $cRiceParam = 1$	Exp-Golomb kth (EGk) $k = 0$
0	000	0	00	1
1	001	10	01	010
2	010	110	100	011
3	011	1110	101	00100
4	100	11110	1100	00101
5	101	111110	1101	00110
6	110	1111110	1110	00111
7	111	1111111	1111	0001000

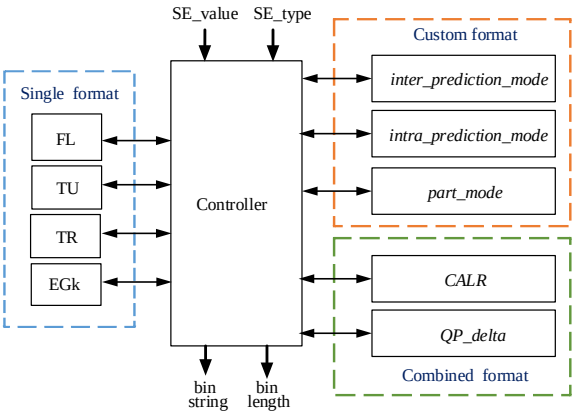

Fig. 3: Block diagram of binarization block.

In combined format, it is performed on a combination of two or more primary methods. For instance, the SE type of '*coeff_abs_level_remaining*' (CALR) and '*cu_qp_delta_abs*' (QP Delta) uses the combination of TR and EGk. In custom, the binarization is performed using a predefined conversion table. Few specific SE types, such as '*inter_pred_mode*', '*intra_pred_mode*', and '*part_mode*' use the custom format [5] and [7].

Table 2 presents the primary BN methods for HEVC and VVC standards. After BN converts the bins, it passes to the next stage (CM/BAE process) as input. These input bins are processed into either regular or bypass bins. The frequency of bin appearance is estimated using the probability model [7], the bins with the probability of at least 0.5 are known as the Most Probable Symbol (MPS), otherwise referred to as Least Probable Symbol (LPS). The regular bins are categorized as the MPS or LPS, which undergoes the CM block [8]. Thus, probability estimation is required for regular bins but not for bypass bins. Without using probability estimation, the bypass bin path considerably reduces coding complexity compared to the regular bin path. Taking advantage of this, it may be extended to process multiple bins simultaneously because it contains no iteration loop, like the regular bin.

The BAE performs an arithmetic coding algorithm and applies the recursive sub-interval division according to the bin type. There are five variables that play a vital role in the algorithm, such as $binValue$ (bin with value), $ivlCurrRange$ (the interval current range), $ivlLow$ (the interval lower bound of this range), $bitsOutstanding$ (value of outstanding bits) and $BinCountsInNal\text{-}Units$ (bin count for network abstraction layer unit). The flow chart of the bypass bins encoding is shown in Fig. 4.

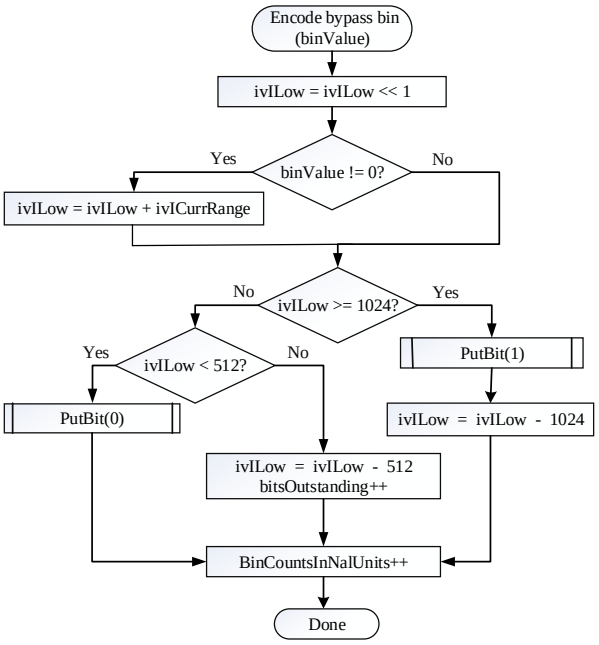

Fig. 4: Flow chart of bypass bin encoding [5].

The $binValue$ is scaled with $ivlLow$ and $ivlCurrRange$. The integer $ivlLow$ is confined to the range [0, 1024) while $ivlCurrRange$ lies in the range [0, 512). The $PutBit$ function is used for the renormalization process. After completion of this process, it obtains 1-bit along with $BinCounts\text{-}InNalUnits$. Then, the BG block processes the output bits and appends them to the output stream. It accumulates the bits while keeping track of the number of $bitsOutstanding$. It also manages the bit packing of the output encoded $bitstream$ so that it may process in the main processor system of the CABAC encoder.

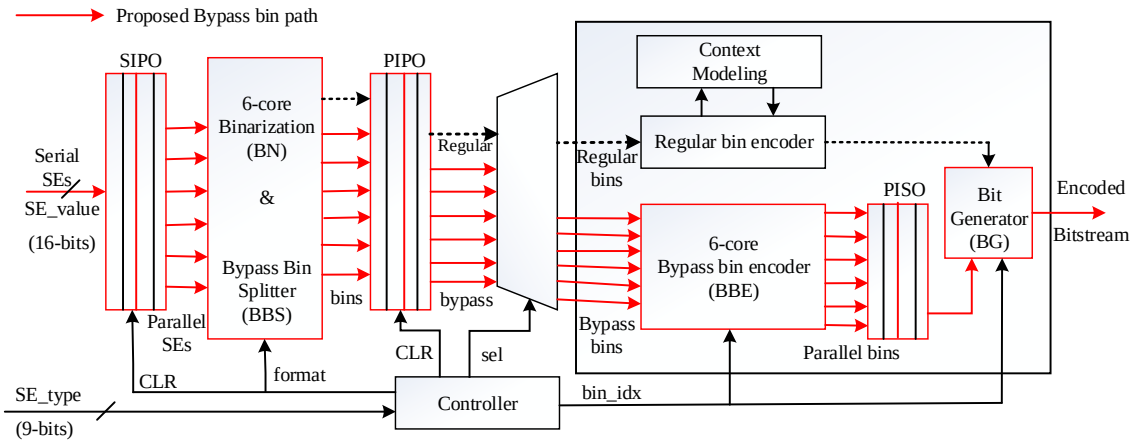

Fig. 5: Proposed bypass bin path architecture.

The trend of designing hardware for CABAC started in 2003, when it was first introduced in the H.264 standard by [6]. Most of the works have focused on developing architectures for BAE. Because it was identified as a bottleneck of the CABAC module. In [8] stated, one of the techniques used to improve the throughput of CABAC is grouping bypass coded bins. Thus, several bypass bins can be processed per clock cycle yielding a significant rise in throughput. One of the initial attempts to process multiple bins was made by [9] by implementing a 2-stage pipelined BAE with dual-symbol encoding for optimized processing of bypass bins, resulting in 1.9–2.3 bins/cycle. In [10] proposed a 4-stage pipelined CABAC with 3-stage pipelined BAE that yielded 1 bin/cycle throughput. In [11] presented three-stage pipelined BAE, one stage for re-normalization, and two stages for bit packing. Three customized sub-modules were used to encode regular bins, bypass bins, and terminate bins. Chen et al. [12] designed a dual-core 6-stage pipelined BAE offering an average throughput of 2.37 bins/cycle.

Recently, semiconductor technologies have significantly reduced latency and enabled a higher clock rate. Few BAE designs utilize multi-core architectures to maximize BAE's throughput [13] and [14]. Pham et al. [13] used different cores to encode each type of bin, processing one bin per cycle. Ramos et al. [14] proposed 4-Binarization Core (BC) architecture for high-throughput and also adopted the AND-based operand isolation for low power. Work of Zhou et al. [15] proposed an ultra-throughput architecture for BAE. It was realized using four parallel-pipelined BAE cores that can encode four regular bins per cycle. Also, they used bypass bin splitting, pre-normalization, hybrid path coverage, and look-ahead range of LPS (rLPS) which considerably reduce BAE's critical path delay. However, the usage of many BAE cores leads to a higher hardware area.

In all these works, different authors reported low-cost solutions, such as balancing performance, area, power and throughput. In the present work, an efficient hardware architecture of the CABAC-Bypass bin path has been proposed. A combination of parallel processing and resource-sharing technique has been used for high-throughput and low hardware area, respectively. The bin splitting scheme for grouping bypass bins and efficient memory management have also been used without effecting in the main CABAC functionality.

3. Proposed Architecture of Bypass Bin Path

An area-efficient high-throughput CABAC encoder architecture is required for encoding UHD video content. The CABAC-bypass bin path hardware architecture has been proposed here to achieve it. Its system-level architecture is shown in Fig. 5. where the bypass bin path is treated as a hardware accelerator. The multiple bins parallel processing architecture has been used to process several bypass bins in one clock cycle.

In the proposed architecture, besides three main blocks, the Bypass Bin Splitter (BBS) and Storage buffers are also used in the data path. The proposed architecture comprises heterogeneous six parallel functional units (6-core binarization and 6-core bypass bin encoder) in bypass bin path; therefore, each clock cycle can handle a maximum of six SEs/bins as input of the binarization/bypass bin encoder at the most. It helps to process the multiple bypass bins (six) in one clock cycle at the bypass bin encoder.

It also consists of several other functional modules, such as three storage buffers, Controller, Multiplexer, and Bit Generator (BG). The three storage buffers are SIPO (Serial In, Parallel Out), PIPO (Parallel In, Par-

allel Out), and PISO (Parallel In, Serial Out). These have been used as per buffer requirements in the proposed bypass bin path.

The controller generates concern control signals to operate all the blocks in the bypass bin path. In the beginning, the serial SEs are buffered using the SIPO and convert them into parallel SEs. These SEs are non-binary, represented with 16 bits (SE_vlaue) and 9 bits (SE_type). Then, the 6-core Binarization (BN) block fetches 6 SEs simultaneously to binarize in parallel and produce respective binstream. This binarized binstream contains the regular and bypass bin together.

On the other hand, the controller generates the control signals to the binarize bins, bypass bins separation, and relative positions. These have been forwarded through a dedicated PIPO buffer. After that, six bypass bins are encoded using the 6-core Bypass Bin Encoder (BBE) and buffered into the PISO. Finally, the BG stage generates the output encoded bitstream as per FSM (Finite State Machine). The following subsections explain each sub-block in detail.

3.1. Architecture of the 6-core BN

In the proposed BN architecture, the controller block has been taken outside of conventional BN architecture and adopted a resource-sharing technique among BN methods. Then, six single-core BN blocks have been operated in parallel. The proposed single-core BN architecture is shown in Fig. 6. It consists of different BN methods, such as FL, TU, TR, and EGk. The inputs given to the BN module are having SE_value (16-bits), SE_type (9-bits), and mode (1-bit).

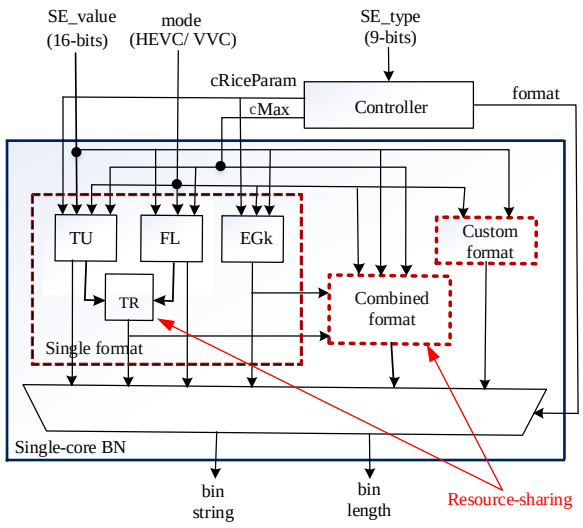

Fig. 6: The proposed single-core binarization architecture.

In the resource-sharing technique, the interdependent TR method has been derived from TU and FL. These BN methods have been utilized to map the non-binary SEs into bins. The four methods (TU, TR, FL, and EGk) have been connected in parallel in a single format. The combined format shares the same resources of a single format. The combined format is a combination of the TR and EGk methods.

The controller module is taken outside of conventional binarization architecture. This controller module activates BN methods based on the given SE_type input. Depending upon SE_type, the 'format' signal from the controller selects the corresponding single BN method and generates the output bin string accordingly. The $cRiceParam$ and $cMax$ have been generated from SE_type, as shown in Fig. 6. Here, the $cRiceParam$ is the control rice parameter, whereas $cMax$ is a variable representing the maximum number of bins allowed to be generated. The mode control signal supports different standards (0 for HEVC and 1 for VVC). The resource-sharing technique has been employed in combined formats that use the same resources available in a single format to achieve low hardware. The output of this BN process is in the form of bin string along with bin length. In [14] and [15], the CABAC throughput requirement is more than 1 Gbin·s^{-1} for 8K UHD. The throughput depends on operating frequency and the number of processed bins per clock cycle given by:

$$\text{Throughput}\,\frac{bin}{s} = \text{Max. Operating Frequency (MHz)} \times \text{No. of Bins Clock Cycle (BPCC)}. \tag{1}$$

In the conventional single-core binarization architecture, approximately 1 bin/cycle is processed [11]. To support high-throughput designs, it is necessary to supply more bins/cycle (typically 4 to 6) at the input of the bypass bin encoder [19] and [20]. Hence, an overestimated binarization architecture has been adopted, which simply uses 6 BN blocks in parallel to process six types of SEs in a single clock cycle. The proposed 6-core binarization architecture is shown in Fig. 7. It supports the mapping of multi-SEs (six) to generate multi-bins (six). However, this parallelization increases the hardware area. Therefore, the resource-sharing technique has been used at each BN core and connected to a common controller block (Fig. 5) to save hardware.

After binarization, the output binstring is a combination of regular and bypass bins. The BBS scheme has been used to split the bypass bins from regular bins using the controller block signal and pass it to the PIPO buffer to balance the bin's processing flow.

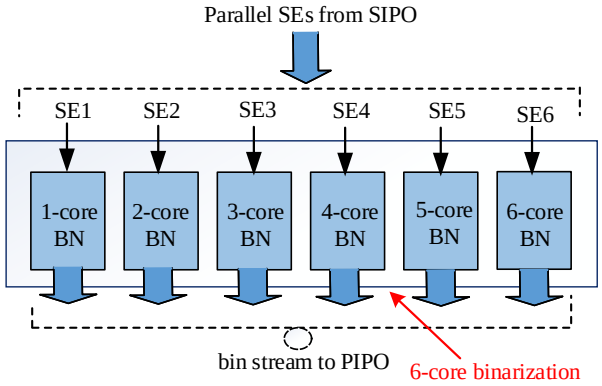

Fig. 7: Proposed 6-core binarization architecture.

3.2. Storage Buffers (SIPO-PIPO-PISO)

The data flow of the proposed architecture (Fig. 5) includes three storage buffers: (i) before BN (SIPO buffer), (ii) after BN, and before BBE (PIPO buffer), and (iii) after BBE (PISO buffer). These three buffers implemented in flip-flops are capable of taking input SEs/bins and output as the multiple bins in each clock cycle. The first buffer, the SIPO, is a 16-bit shift register that consists of 16 Flip-Flops (FFs). It takes 16 clock cycles (equal to the number of FF stages) to form a single SE. The serial to parallel SE processing has been taken care of by this synchronous SIPO buffer. The controller has been connected to the SIPO buffer, which controls the parallel output SEs. The second buffer is a 16-bit PIPO buffer which is used to place the binarized output. The PIPO buffer has been placed between the 6-core BN and 6-core BBE to balance data flow in the bin processing path. The third buffer PISO, has been used after the 6-core BBE output, which converts parallel to serial data before bins pass to the BG block.

3.3. Architecture of the 6-core BBE

The Bypass bin encoding engine is based on six variables at the input, such as (i) $binValue$, (ii) $ivlLow$, (iii) $ivlCurrRange$, (iv) $bitsOutstanding$, (v) $BinCountsInNalUnit$, and (vi) $PutBit$. There are four variables obtained at the output of this process, namely: (i) $ivlLow$, (ii) $bitsOutstanding$, (iii) $BinCountsInNalUnit$, and (iv) $PutBit$. The proposed architecture of a single-core BBE engine has been shown in Fig. 8.

According to the HEVC standard [5], the bypass bins account for a significant portion of the total bins. The bypass bin encoder is a simplified version of the regular bin encoder. As bypass bins having the LPS of 50 %, these bins have not been used for CM and probability

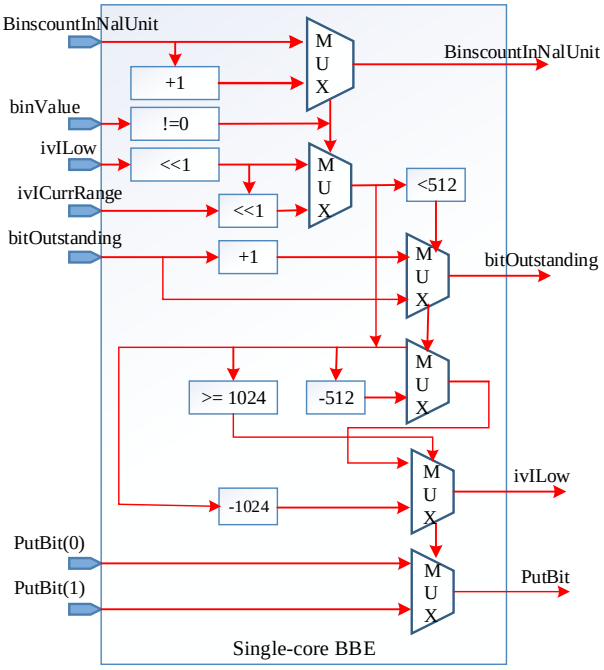

Fig. 8: Proposed Single-core bypass bins encoder.

estimation [7] and [16]. Hence, the multiple bins are processed per clock cycle. The six bins are processed in parallel to improve the BPCC of the CABAC encoder, as shown in Fig. 9.

The 6-core bypass bin encoder receives up to six consecutive bypass bins (binValue0 to binValue5) and processes in a single processing cycle using six parallel single-core BBE processing units. An intermediate output of each processing unit has been coupled to inputs of the subsequent processing unit ($ivlLow$ and $ivlRange$). Hence, six input bins have been fed in parallel during a single clock cycle. A bypass bin resolving unit accepts the intermediate output of the processing units ($bitsOutstanding$) and generates combined bins from each core output bin. The output of this 6-core BBE is sent to the PISO buffer, which converts parallel bins into serial.

3.4. Bit Generator (BG)

The output bins from the PISO buffer have been put into bit generation block to form output encoded bitstream. The block diagram and state diagram of the bitstream generator have been referred from [17] and [18] and reproduced here in Fig. 10 and Fig. 11, respectively.

The bit generator operates as a FSM. There are five FSM states: (i) idle (IDLE), (ii) output standing (OUT_STD), (iii) output count (OUT_CNT), (iv) next state (NEXT), and (v) end of the stream (EOS). The string of parsed bins cannot be directly

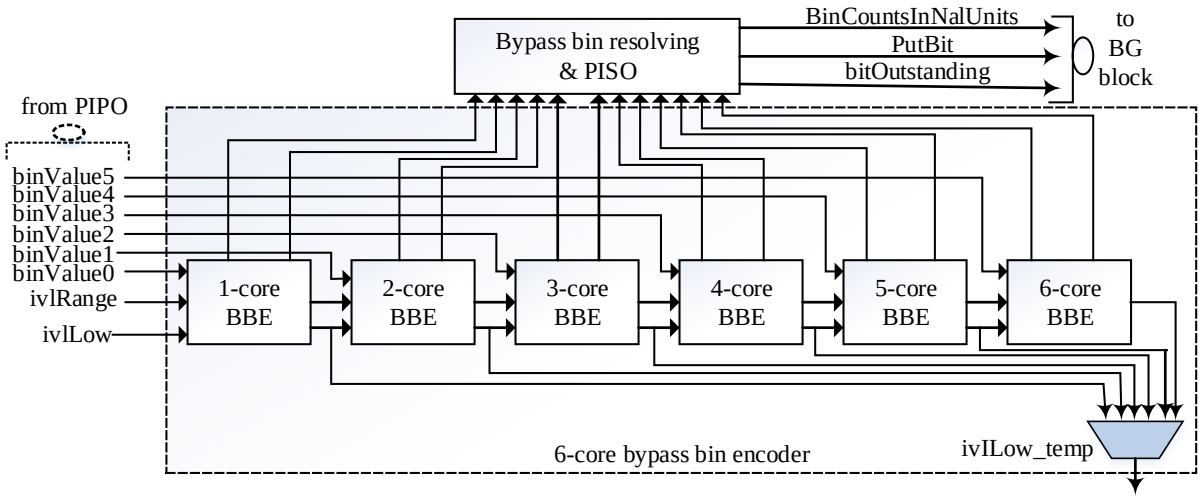

Fig. 9: Proposed 6-core Bypass Bin Encoder (BBE).

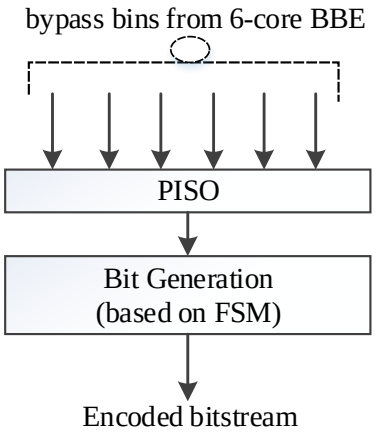

Fig. 10: The data flow of the Bit Generator.

sent to the output stream since the polarity of the potential outstanding bits is yet to be resolved [18]. An intermediate buffer is required to accumulate these parsed bins and issue them to the input PISO when the polarity of the outstanding bits is resolved.

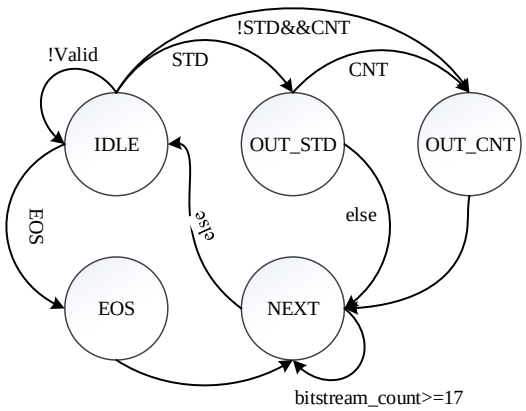

Fig. 11: State diagram of Bit Generator [17].

The BG outputs a bitstream through an information of the current bin. It processes the *bitOutstanding* ones to generate a bitstream in a sequence of bits.

4. Experimental Test Setup, FPGA and ASIC Implementation

The proposed architecture has been modeled in Verilog Hardware Description Language (HDL), simulated with XILINX ISE 14.7, and implemented on NEXYS4 DDR Board [23]. The test-bench has been created, and functional verification has been performed. The design has been verified with an on-chip debugging tool (chipscope pro logic analyzer). The experimental test setup, verification process (Reference software and Proposed hardware) have been shown in Fig. 12. The figure also shows the system-level structure of interfacing between hardware, software, including tools and environment.

Further, to analyze the proposed design's correctness, Register Transfer Level (RTL) simulation has been performed using recommended standard test videos. To assess the performance of design, three test video sequences (Basketball, PeopleOnStreet, and Traffic) have been taken [24]. Two different HEVC configurations: (i) Low Delay (LD), (ii) Random Access (RA) and two Quantization Parameters (QPs) of 22 and 37 (minimum and maximum recommended values), have been considered for testing [20]. These video sequences having UHD resolution. Table 3 shows the test results of the proposed design. The proposed design has achieved a maximum throughput of an average 6 bins/cycle (QP@22) as it has the ability to process 6 SEs in a clock cycle. The throughput (Average BPCC) of our architecture has been analyzed in

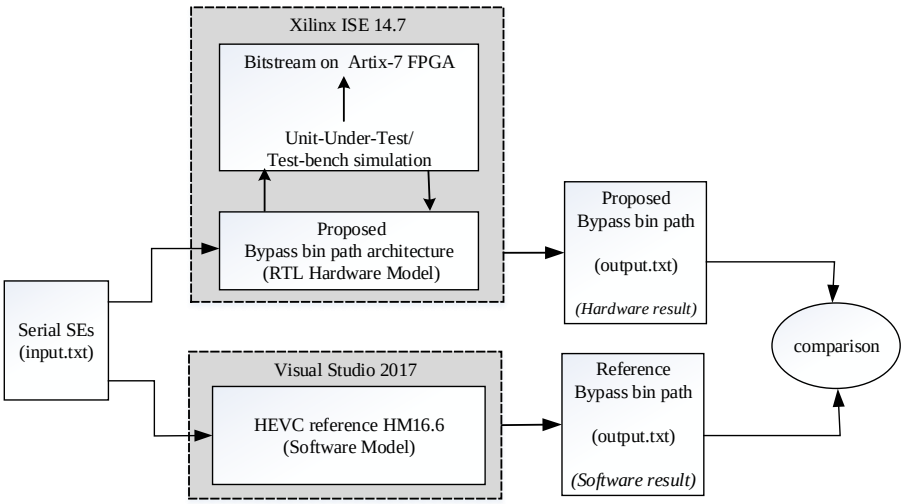

Fig. 12: An illustration of the experimental test setup.

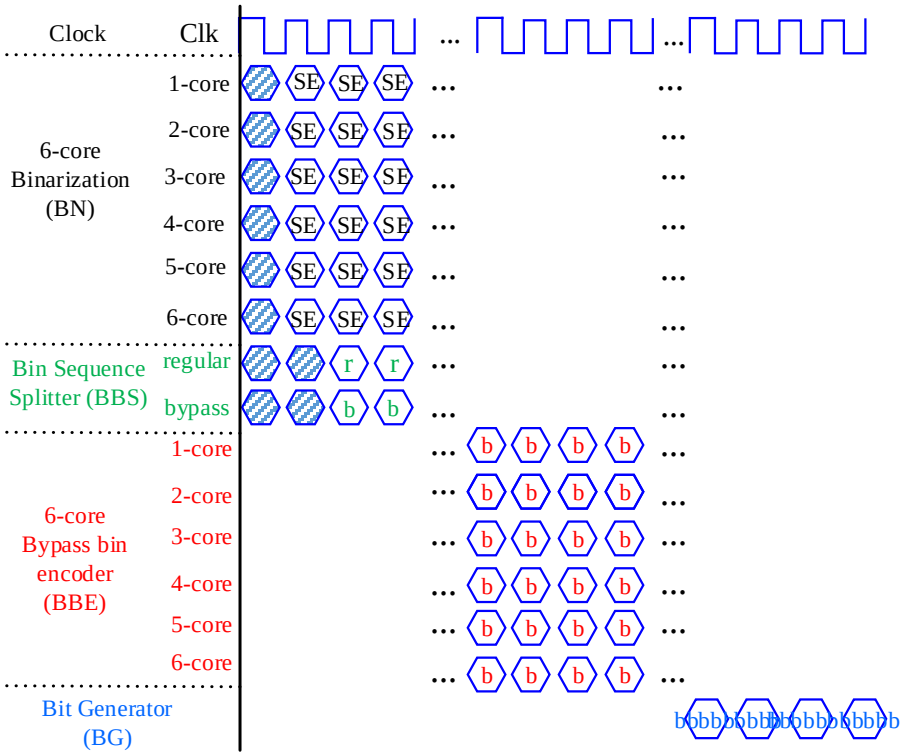

Fig. 13: The timing diagram of the bypass bin processing. A striped block denotes an idle state, whereas a white block represents the working state. The symbols in white blocks are specified as a SE, r (regular bin), and b (bypass bin).

absolute and relative terms of the test video sequences. Nevertheless, Fig. 13 presents the timing diagram of the proposed design in terms of BPCC at the CABAC system level. The process of SEs to bins generation has also been shown.

In the present implementation, the bypass bin has been finished within a single cycle rather than the original three cycles by proper memory arrangements, as illustrated in Fig. 14. The quantized and transform coefficients occupy (Fig. 1)

Tab. 3: Performance of proposed design.

Test sequence	Configuration	Bypass bins per clock cycle	
		QP@22	QP@37
Basketball	LD	5	5
	RA	7	6
PeopleOnStreet	LD	6	5
	RA	5	6
Traffic	LD	6	5
	RA	7	5
Average BPCC		**6.00**	**5.33**

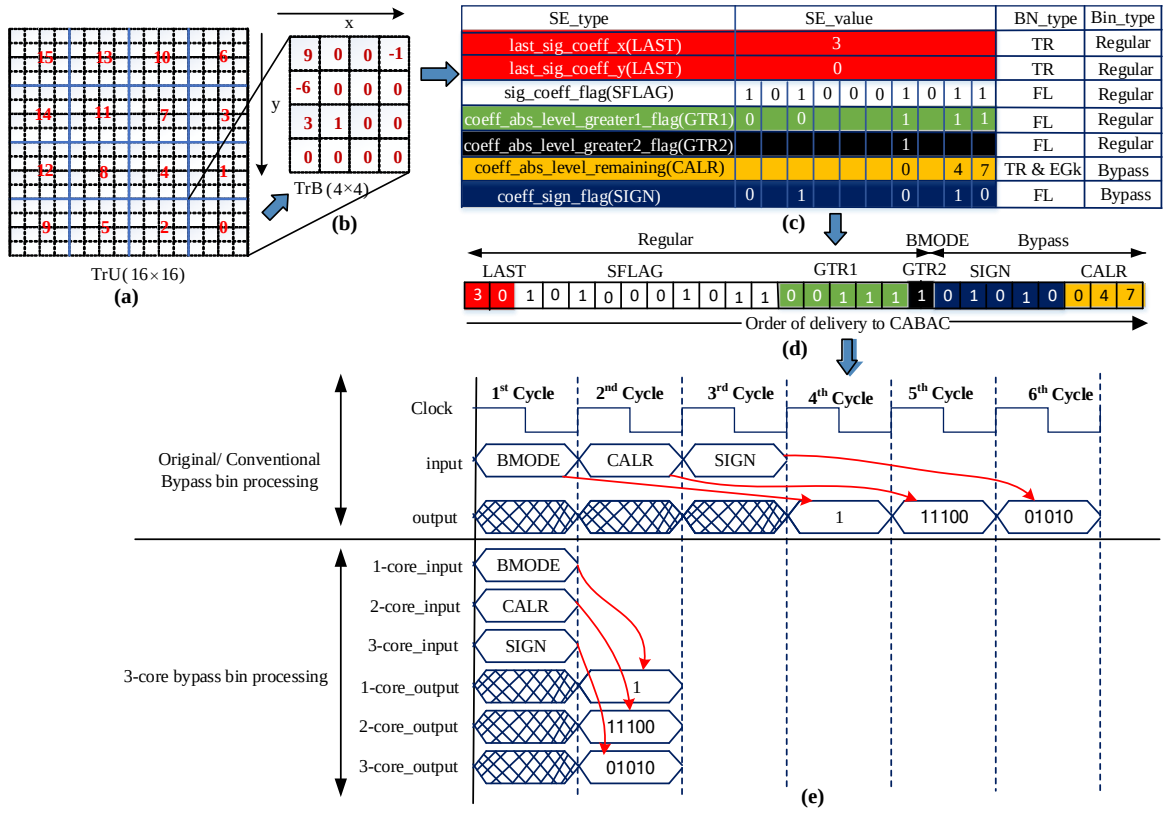

Fig. 14: The BPCC of TrB 4×4 size (a to e). A striped block denotes an idle state, whereas a white block represents the working state.

a notable amount of SEs, 75 % on the average and 94 % in the worst-case [7] and [21]. This CABAC input processing data, known as Transform Block (TrB) of size 4×4 contains three types of SEs [5], such as (i) *cabac_bypass_alignment_enabled_flag* (BMODE), (ii) *coeff_sign_flag* (SIGN), and (iii) *coeff_abs_level_remaining* (CALR). These are processed through the bypass bin path, as shown in Fig. 14.

Figure 14(a) shows a quantized Transform Unit (TrU) of size 16×16. The diagonal scan has been applied to divide TrU into non-overlapped 4×4 of TrB, as shown in Fig. 14(b). The table contains a diagonal scan extracted data of 4×4, such as *SE_type*, *SE_vlaue*, Binarization method (*BN_type*), and type of bin processing (*Bin_type*) as per HEVC standard [5] are given in Fig. 14(c). In Fig. 14(d), the conversion of each 4×4 TrB to the 1D array of 16 consecutive coefficients for the CABAC processing order is presented. Finally, the number of clock cycles required to process the bypass bins is shown in Fig. 14(e). These three SEs are processed in parallel by the proposed 6-core bypass bin encoder (in this case, only 3-cores utilized) and with the help of SIPO-PIPO-PISO storage buffer arrangements which produces the encoded bitstream in a single cycle. As a result, it is verified that the process of the current bin is improved on the completion of the last bin generation to save the clock cycles.

The proposed architecture has been synthesized on a 27 nm ARTIX-7 FPGA device (NEXYS-4 DDR) which consumes 991 Slice Registers and 719 Slice LUTs. The FPGA implementation also uses 12 SRLs (for storage buffers). Table 4 summarizes the slice registers, Slice LUTs, and SRLs for each module. The maximum operating frequency of storage buffer, 6-core BN, and 6-core BBE are 264 MHz, 187 MHz, and 320 MHz. The overall maximum operating frequency of the proposed design is 112 MHz after FPGA post place and route (P&R) stage.

Tab. 4: FPGA implementation results.

Resource used	Storage buffers	6-core BN	6-core BBE	Total
Slice registers	376	212	403	991
No. of Slice LUTs	287	247	185	719
LUT-FF fully	-	2052	372	2424
SRLs	12	-	-	12

The ASIC implementation has been done using 65 nm CMOS technology. The synthesis netlist has been generated by using CADENCE GENUS 16.6. The proposed design has been synthesized at each core

stage. Their respective results have been tabulated in terms of Average BPCC, maximum operating frequency, achieved maximum throughput, and consumed hardware area (number of gates), as shown in Tab. 5. The design provides an average of 1 BPCC for single-core and 6 BPCC for 6-core architecture.

Tab. 5: ASIC Synthesis results of core stages.

Core stages (BN+ BBE)	Max. SEs/ clock cycle	Avg. BPCC	Max. freq. (MHz)	Max. throughput (Gbin·s^{-1})	Hardware area (No. of gates)
1-core	1	1	720	0.72	1458
2-core	2	2	467	0.93	2916
3-core	3	3	318	0.95	4354
4-core	4	4	243	0.97	5832
5-core	5	5	218	1.09	7290
6-core	6	6	210	1.26	8758

The ASIC post P&R performance results of the 6-core bypass bin path are presented in Tab. 6. The total gate count is 8,758 gates including storage buffers with a maximum operating frequency of 210 MHz. Table 6 also shows the ASIC implementation results of the proposed design at system level in terms of the minimum and maximum value of SEs/BPCC, Maximum operating frequency, Maximum throughput, and total hardware area (number of gate count). It is verified that the proposed design encodes on an average of 6 BPCC at the 6-core stage (Tab. 3).

Tab. 6: Proposed bypass bin path ASIC Implementation results.

CMOS technology (nm)		65
Max. operating frequency (MHz)		210
SEs per clock cycle (min–max)		1–6
BPCC (min–max)		5–7
Average BPCC		6
Max. throughput (Gbin·s^{-1})		1.26
Gate counts module-wise (No. of gates)	6-core BN	2,683
	6-core BBE	4,443
	Bit generator & Storage buffers	1,632
	Total gate count	8,758

CADENCE INNOVUS 16.7 and VIRTUOSO 6.2 were used for layout implementation. The GDSII layout of the proposed design has positive slack. The proposed design fulfills the requirement of no timing errors. The post layout design has passed the DRC/LVS checks. The RC extraction has also been done for physical verification. The final chip layout occupies 2.01 mm^2.

5. Results and Discussion

The main objective of the present work is to design and implement a CABAC having high-throughput and low hardware area, that have contrary to requirements [8]. The bypass bins occupy a significant portion of the total bins in the CABAC [7]. Therefore, overall CABAC throughput can be notably improved by processing multiple bypass bins per clock cycle. To support high-throughput UHD, it is necessary to supply typically 4 to 6 at the input of the bypass bin encoder [18]. Here, six separate cores with a low area technique (resource-sharing) have been used to generate up to 6 bypass bins to increase throughput and reduce hardware area. When bypass bins are grouped and encoded in parallel, it enhances the CABAC throughput [7] and [15]. The SEs of coefficient level related to the bypass bins occupy up to 25.6 % of total bins in CABAC [14]. The proposed architecture utilizes the BBS (Fig. 14(c), Fig. 14(d) and Fig. 14(e)) along with grouping bypass bins based on SE types, as shown in Tab. 8.

By using this approach, it made possible to process six bypass bins in one clock cycle. Thus, multiple bypass bins have been processed in the single-cycle, resulting in high-throughput than non-grouped SEs having frequent switching between bypass and regular bins. Moreover, an increasing the number of processed bins per clock cycle and the clock frequency have also been recorded in the present work. The data of Tab. 5 are plotted and shown in Fig. 15.

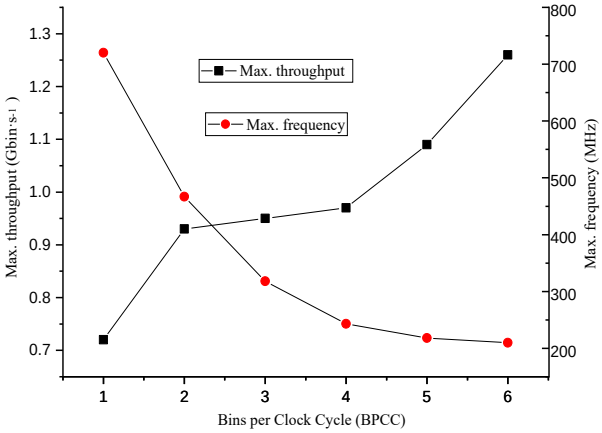

Fig. 15: Relationship between BPCC, Throughput, and Frequency.

The throughput improvement of bypass bin processing lies with the increment of BPCC (Eq. (1) but results in a lower operating frequency, as clear from Fig. 15. It has been observed that due to an increment in the processing path delay that occurred before and after the BN process stages. Figure 16 shows the timing diagram of the SIPO buffer.

Each SE (16-bits) requires 16 clock cycles by a single-core BN at the SIPO buffer stage. Besides it, the PIPO and PISO buffer delays also cause a reduction in the operating frequency of the design. These delays increase along with the depth of parallel processing levels. As a result, a trade-off exists between throughput and performance (i.e., operating frequency) over BPCC.

Tab. 7: Comparison with existing works.

Reference	[13] 2014	[14] 2016	[15] 2015	[16] 2013	[17] 2015	[18] 2017	[19] 2018	[20] 2018	[21] 2019	[22] 2020	This work
Max. SEs per clock cycle	*	*	4	*	*	*	*	*	4	3.5	6
Avg. BPCC	1	4	4.37	1.18	1.92	1.4	4.94	4.56	4.5	3.05	6
Max. frequency (MHz)	180	280	420	357	136	810	537	264	668	500	210
Max. throughput (Gbin·s^{-1})	0.18	1.12	1.83	0.43	0.26	1.13	2.65	1.20	2.67	1.52	1.26
Gate count (Kgates)	3.69	9.95	64.10	48.94	41.6	2.20	33	14.6	3.67	9.45	8.76
Area-efficiency (Gbins/ Kgate)	0.48	0.11	0.02	0.008	0.006	0.51	0.08	0.082	0.727	0.160	0.143
Memory type	*	Registers	PIPO	PISO and RAM	FIFO	Barrel shifter	PIPO	Registers	Registers	TB Memory	Storage buffers
Technology process (nm)	180	45	90	130	180	45	65	65	65	45	65
Implemented CABAC block	BAE	BAE	Full CABAC	Full CABAC	Bypass bin	Bypass bin	Bypass bin	Bypass bin	Residual SE generation	SE generate+ binarization	Bypass bin
Supported resolution	2K	UHD	UHD	2K	*	UHD	UHD	UHD	2K	UHD	UHD
Video support	HEVC	HEVC	HEVC	HEVC	HEVC	HEVC	HEVC	HEVC	HEVC	HEVC	HEVC/VVC
Design strategy	*	4-core BAE structure	Reduced critical path delay	Parallel process	4-stage pipeline	4-stage pipeline	8-stage pipeline BAE	*	Multi residual SE treatment (MRSET)	Combined SE binarization hardware	Parallel & resource-sharing

*Not Reported

Tab. 8: The group of SEs for the bypass bin process.

No. of Groups	**Syntax Element type** (SE_type)	**No. of SEs**
Group1	Motion vector difference	2
Group2	Coefficient level and Sign	2
Group3	Reference index	2
Group4	QP Delta	5
Group5	Reminder of intra prediction mode	4

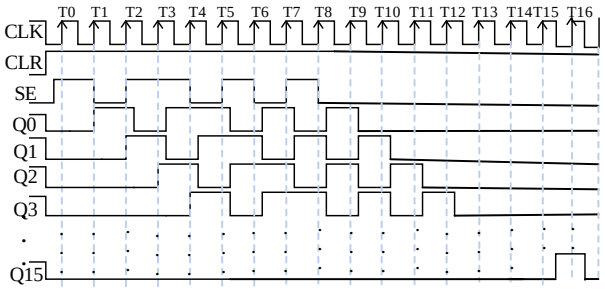

Fig. 16: The timing diagram of SIPO Buffer.

Furthermore, to measure system throughput, the proposed design has been tested using the standard video sequences shown in Tab. 3. As per results, the QP@22 requires a higher average BPCC (6.00) than QP@37 (5.33) due to an increase in non-zero residue data at a minimum quantization step (QP@22). Furthermore, the proposed architecture also provides a flexible choice of selecting the appropriate number of core stages, as shown in Tab. 9 with supported resolutions. The throughput requirement for UHD (4K and above) of 1 Gbin·s^{-1} is supported by 5-core and 6-core of the proposed architecture.

The comparison of present work with existing works has been presented in Tab. 7. Several authors have reported the architectural implementation with different

Tab. 9: Proposed scalable architecture. The design cores, achieved throughput and the corresponding supported resolution.

Core stages (BN+BBE)	**Max. throughput (Gbin·s^{-1})**	**Supported resolution***
1-core	0.72	<1K
2-core	0.93	1K
3-core	0.95	2K/4K
4-core	0.97	4K
5-core	1.09	4K/8K
6-core	1.26	8K and Beyond

*<1K - HD, 1K - Full HD, 2K and Above - UHD

strategies at block level [13], [14], [17], [18], [19], [20], [21] and [22], while few implemented the full CABAC [15] and [16]. All the works reported are implemented in different technologies and platforms. Thus, the comparison of the proposed work with them will not give an accurate estimate.

However, a comparison based on processed SEs per clock cycle and BPCC has been considered, which is independent of technology nodes and platforms. The proposed work has achieved higher SEs per clock cycle and Average BPCC among all. This is possible due to massive parallelism using intermittent storage buffers. Some works achieved higher throughput with large area overhead [14], [15], [19], [20] and [22], while few consume moderate and even lower area [13], [18] and [21]. A fair comparison has been made with the works of [19] and [20] because these designs use the same technology node (65 nm) and also use the bypass bin process. The proposed design achieves 6 BPCC compared to approximately 5 BPCC by [19] and [20] while occupying around half the gate area with less clock frequency, as shown in Fig. 17.

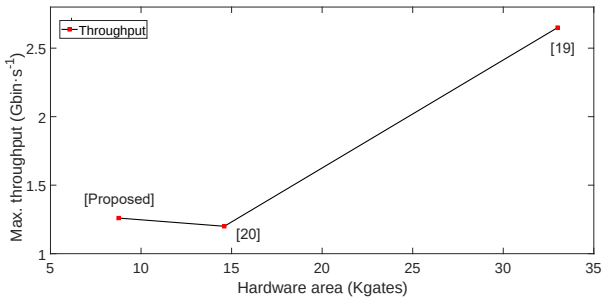

Fig. 17: Comparison in terms of throughput and area.

In case of area-efficiency (calculated by Max throughput to Gate count ratio, Gbins/Kgate) among the bypass bin works, the work [18] is only the most efficient design than our proposal, which are achieved 0.51 and 0.143, respectively. Furthermore, the result of our architecture has been analyzed in absolute and relative terms, compared with the CABAC's similar implementation.

When comparing with [13], [18], and [21], although the gate count of the proposed work is larger, the BPCC is higher. In comparison with other block-level designs [14], [20], and [22], the proposed work requires less hardware area. It has been possible due to the resource-sharing technique, paralleling at the binarization block level and the bypass bin encoder. The parallelizing effects on the datapath are also considered using the storage buffers.

To summarize, the proposed design caters to both processing capability (high-throughput) and low hardware area (area-efficient). It represents novelty by scalable parallel architecture with resource-sharing that delivers six bypass bins per clock cycle. Hence, it exhibits high-throughput in terms of the BPCC and a low amount of gate count.

In the HEVC standard, maximum bitrate is defined as 800 Mbit·s^{-1} for 8K UHD Television applications that corresponds to a bin rate of approximately 1 Gbin·s^{-1} [5], [14], and [15]. The proposed 6-core bypass bin encoder achieved a throughput of 1.26 Gbin·s^{-1}. So, it is suitable for the 8K UHD TV applications.

6. Conclusion

The Binary Arithmetic Encoder is the most crucial component of CABAC because of its serial bin based processing and inherent data dependencies. This work proposed a design and implementation of high-throughput and area-efficient architecture for the bypass bin processing employing six independent bypass bin encoders with parallel processing. Using bypass bin splitter, it enhanced the parallel processing of bypass bins. The proposed design also considers the binarization process by processing 6 SEs/cycle and efficient memory management for maintaining the processing path.

The proposed architecture increases the throughput by 6 bypass bins per clock cycle than the existing architectures, capable of processing between 1–5 bins/cycle. The architecture has been successfully implemented using 65 nm technology library with a maximum operating frequency of 210 MHz. The design can process 1.26 Gbin·s^{-1}, which meets the requirement for processing of 8K resolution video. The resource-sharing technique employed in parallel processing architecture has significantly saved the hardware area compared to other reported architectures. The throughput result depends on the architecture but not on the technology used for implementation. It enhances the whole performance of the CABAC encoder, which can be applied to multimedia devices and systems for UHD and beyond. This architecture also supports multi-standard (HEVC/VVC) video coding.

Acknowledgment

The authors acknowledge the resources utilized from the VLSI Laboratory of the Department of Electronics and Communication Engineering (ECE), MNNIT Allahabad, Prayagraj-211004, UP, India. The Ministry of Electronics and Information Technology (MeitY), Government of India, funds the project under Special Manpower Development Program - Chip to System Design (SMDP-C2SD).

Author Contributions

N.M. has contributed to the design and implementation, the analysis of the results, and the first draft of the manuscript. S.K.G. has provided critical comments during design and implementation. S.K.G. and V.B. have supervised the work, did the review, and necessary editing in the final version of the article.

References

[1] UHRINA, M., J. FRNDA, L. SEVCIK and M. VACULIK. Impact of H.264/AVC and H.265/HEVC Compression Standards on the Video Quality for 4K Resolution. *Advances in Electrical and Electronic Engineering*. 2014, vol. 12, iss. 4, pp. 368–376. ISSN 1804-3119. DOI: 10.15598/aeee.v12i4.1216.

[2] WIEGAND, T., G. J. SULLIVAN, G. BJONTEGAARD and A. LUTHRA. Impact of H.264/AVC and H.265/HEVC Compression Standards on the Video Quality for 4K Resolution. *IEEE Transactions on Circuits and Systems for Video Technology*. 2003, vol. 13, iss. 7, pp. 560–576. ISSN 1558-2205. DOI: 10.1109/TCSVT.2003.815165.

[3] SULLIVAN, G. J., J.-R. OHM, W.-J. HAN and T. WIEGAND. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Transactions on Circuits and Systems for Video Technology*. 2012, vol. 22, iss. 12, pp. 1649–1668. ISSN 1558-2205. DOI: 10.1109/TCSVT.2012.2221191.

[4] CHEN, J., M. KARCZEWICZ, Y.-W. HUANG, K. CHOI, J.-R. OHM and G. J. SULLIVAN. The Joint Exploration Model (JEM) for Video Compression With Capability Beyond HEVC. *IEEE Transactions on Circuits and Systems for Video Technology*. 2020, vol. 30, iss. 5, pp. 1208–1225. ISSN 1558-2205. DOI: 10.1109/TCSVT.2019.2945830.

[5] H.265/ISO/IEC 23008-2 HEVC. *High efficiency video coding*. Geneva: ITU-T, 2019.

[6] MARPE, D., H. SCHWARZ and T. WIEGAND. Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard. *IEEE Transactions on Circuits and Systems for Video Technology*. 2003, vol. 13, iss. 7, pp. 620–636. ISSN 1558-2205. DOI: 10.1109/TCSVT.2003.815173.

[7] SZE, V., M. BUDAGAVI and G. J. SULLIVAN. *High Efficiency Video Coding (HEVC): Algorithms and Architectures*. 1st ed. Cham: Springer, 2014. ISBN 978-3-319-06895-4.

[8] SZE. V. and M. BUDAGAVI. High Throughput CABAC Entropy Coding in HEVC. *IEEE Transactions on Circuits and Systems for Video Technology*. 2012, vol. 22, iss. 12, pp. 1778–1791. ISSN 1558-2205. DOI: 10.1109/TCSVT.2012.2221526.

[9] OSORIO, R. R. and J. DBRUGUERA. High-Throughput Architecture for H.264/AVC CABAC Compression System. *IEEE Transactions on Circuits and Systems for Video Technology*. 2006, vol. 16, iss. 11, pp. 1376–1384. ISSN 1558-2205. DOI: 10.1109/TCSVT.2006.883508.

[10] ZHENG, W., D.-X. LI, B. SHI, H.-S. LE and M. ZHANG. Efficient pipelined CABAC encoding architecture. *IEEE Transactions on Consumer Electronics*. 2008, vol. 54, iss. 2, pp. 681–686. ISSN 1558-4127. DOI: 10.1109/TCE.2008.4560147.

[11] TIAN, X., T. M LE and Y. LIAN. *Entropy Coders of the H.264/AVC Standard: Algorithms and VLSI Architectures*. 1st ed. Heidelberg: Springer, 2011. ISBN 978-3-642-14703-6.

[12] CHEN, J.-W., L.-C. WU, P.-S. LIU and Y.-L. LIN. A high-throughput fully hardwired CABAC encoder for QFHD H.264/AVC main profile video. *IEEE Transactions on Consumer Electronics*. 2010, vol. 56, iss. 4, pp. 2529–2536. ISSN 1558-4127. DOI: 10.1109/TCE.2010.5681137.

[13] PHAM, D. H., J. MOON, D. KIM and S. LEE. Hardware Implementation of HEVC CABAC Binary Arithmetic Encoder. *Journal of Institute of Korean Electrical and Electronics Engineers*. 2014, vol. 18, iss. 4, pp. 630–635. ISSN 2288-243. DOI: 10.7471/IKEEE.2014.18.4.630.

[14] RAMOS, F. L. L., J. GOEBEL, B. ZATT, M. PORTO and S. BAMPI. Low-power hardware design for the HEVC Binary Arithmetic Encoder targeting 8K videos. In: *29th Symposium on Integrated Circuits and Systems Design (SBCCI)*. Belo Horizonte: IEEE, 2016, pp. 1–6. ISBN 978-1-5090-2736-1. DOI: 10.1109/SBCCI.2016.7724044.

[15] ZHOU, D., J. ZHOU, W. FEI and S. GOTO. Ultra-High-Throughput VLSI Architecture of H.265/HEVC CABAC Encoder for UHDTV Applications. *IEEE Transactions on Circuits and Systems for Video Technology*. 2015, vol. 25, iss. 3, pp. 497–507. ISSN 1558-2205. DOI: 10.1109/TCSVT.2014.2337572.

[16] PENG, B., D. DING, X. ZHU and L. YU. A hardware CABAC encoder for HEVC. In: *IEEE International Symposium on Circuits and Systems (ISCAS)*. Beijing: IEEE, 2013, pp. 1372–1375. ISBN 978-1-4673-5762-3. DOI: 10.1109/ISCAS.2013.6572110.

[17] KIM, D., J. MOON and S. LEE. Design of HEVC CABAC Encoder With Parallel Processing of Bypass Bins. *Journal of Institute of Korean Electrical and Electronics Engineers*. 2015, vol. 19, iss. 4, pp. 583–589. ISSN 2288-243X. DOI: 10.7471/IKEEE.2015.19.4.583.

[18] NGUYEN, Q.-L, D.-L. TRAN, D.-H. BUI, D.-T. MAI and X.-T. Tran. Efficient Binary Arithmetic Encoder for HEVC with multiple bypass

bin processing. In: *7th International Conference on Integrated Circuits, Design, and Verification (ICDV)*. Hanoi: IEEE, 2017, pp. 82–87. ISBN 978-1-5386-3377-9. DOI: 10.1109/ICDV.2017.8188644.

[19] RAMOS, F. L. L., B. ZATT, M. S. PORTO and S. BAMPI. High-Throughput Binary Arithmetic Encoder using Multiple-Bypass Bins Processing for HEVC CABAC. In: *IEEE International Symposium on Circuits and Systems (ISCAS)*. Florence: IEEE, 2018, pp. 1–5. ISBN 978-1-5386-4881-0. DOI: 10.1109/ISCAS.2018.8350885.

[20] RAMOS, F. L. L., B. ZATT, M. S. PORTO and S. BAMPI. Novel Multiple Bypass Bin Scheme and Low-power Approach for HEVC CABAC Binary Arithmetic Encoder. *Journal of Integrated Circuits and Systems*. 2018, vol. 13, iss. 3, pp. 1872–0234. ISSN 1872-0234. DOI: 10.29292/jics.v13i3.3.

[21] RAMOS, F. L. L., A. V. P. SAGGIORATO, B. ZATT, M. PORTO and S. BAMPI. Residual Syntax Elements Analysis and Design Targeting High-Throughput HEVC CABAC. *IEEE Transactions on Circuits and Systems I: Regular Papers*. 2019, vol. 67, iss. 2, pp. 475–488. ISSN 1558-0806. DOI: 10.1109/TCSI.2019.2932891.

[22] TRAN, D.-L., X.-T. TRAN, D.-H. BUI and C.-K. PHAM. An Efficient Hardware Implementation of Residual Data Binarization in HEVC CABAC Encoder. *Electronics*. 2020, vol. 9, iss. 4, pp. 1–12. ISSN 2079-9292. DOI: 10.3390/electronics9040684.

[23] Digilent Nexys4 DDR Board. In: *XILINX* [online]. 2021. Available at: `https://www.xilinx.com/support/university/boards-portfolio/xup-boards/DigilentNexys4DDR.html`.

[24] BOSSEN, F. Common Test Conditions and Software Reference (JCTVC-L1100). In: *Fraunhofer-Institut für Nachrichtentechnik* [online]. 2021. Available at: `https://hevc.hhi.fraunhofer.de/`

About Authors

Nagaraju MAMIDI (Corresponding author) was born in Telangana, India. He received his B.Tech degree in Electrical & Electronics Engineering and M.Tech. in VLSI System Design in 2007 and 2012, respectively, from Jawaharlal Nehru Technical University-Hyderabad, India. He has six years of industrial experience and presently pursuing his Ph.D. degree in Electronics & Communication Engineering Department (ECED), Motilal Nehru National Institute of Technology Allahabad (MNNIT Allahabad), Prayagraj, India. His research interests include High-Performance VLSI architectures and Low-Power Video codecs (H.264, HEVC, and VVC) for UHD Applications.

Santosh Kumar GUPTA was born in Uttar Pradesh, India. He received his B.E. degree in Electronics & Telecommunication Engineering from Govind Ballabh Pant Engineering College, Pauri, Uttarakhand in 2000, M.Tech. in Communication from Indian Institute of Technology Bombay (IIT Bombay), Mumbai, India in 2002, and Ph.D. from National Institute of Technology Silchar (NIT Silchar), Assam, India in 2014. Presently, he is working as an Associate Professor in ECED at MNNIT Allahabad, Prayagraj, India. His research interests include VLSI Design, Simulation, and Modeling of semiconductor devices. He is an author/co-author several research papers in international, national journals and conferences.

Vijaya BHADAURIA was born in Uttar Pradesh, India. She received her B.E. degree in Electrical Engineering and M.E. Degree in Control and Instrumentation in 1984 and 1986, respectively, from MNREC (Now, MNNIT), Allahabad, India. She joined as a faculty member in the ECED of MNREC, Allahabad, in 1986. She obtained her Ph.D. degree in 2012 from the ECED, MNNIT, Allahabad, Prayagraj, India, and currently has been serving as a Professor in the same institute. Her field of interest includes Digital and Analog VLSI IC Design. She is an author/co-author several research papers in international journals and conferences.