# Design of Multi-Bin-Per-Cycle HEVC CABAC Encoder with Multiple Bypass Bin Engines

Doohwan Kim[1], Sohyun Kim[2], Jeonhak Moon[3], Seongsoo Lee[4]
School of Electronic Engineering, Soongsil University, Seoul, Korea
E-mail: doohwan@ssu.ac.kr[1], thgus0624@ssu.ac.kr[2], munggea@gmail.com[3], sslee@ssu.ac.kr[4]

**Abstract:** CABAC (Context Adaptive Binary Arithmetic Coding) is high-compression-ratio entropy encoding in HEVC. However, it is difficult to implement fast CABAC encoder, since each bin should be serially processed due to data dependencies of bins. There are two types of bins, i.e. regular bin and bypass bin. Processing time of bypass bin is shorter than that of regular bin, so it is possible to cascade bypass bin engines for multi-bin-per-cycle processing. Number of bypass bin engines is carefully chosen based on thorough analysis. The implemented CABAC encoder can process up to 1.74 bins per cycle.

*Keywords*—**HEVC, CABAC, multi-bin, bypass**

## 1. Introduction

CABAC encoder architecture consists of three main blocks, i.e. binarizer (BI), context modeler (CM), and binary arithmetic encoder (BAE). BI converts syntax element (SE) to bin string based on SE type and coding tree unit (CTU) information. CM splits binstring into bins and determines encoding type of each bin. It also determines context model based on the probability of bins when the encoding type is regular. In this case, context model is updated in each bin, since encoding of next bin often refers to this updated context model. BAE converts bin to bitstream using its context model or fixed probability. This general architecture is shown in Figure 1 [1]-[4].

In general, multiple processing engines are exploited in parallel to increase hardware performance when there is no data dependency. However CABAC has large data dependency, especially in regular mode. It means that processing of bins should be performed serially. In regular mode, each bin updates the context model, range and range's offset (*low*), and next bin has to refer to these data. So its computation is very complex, and the critical path is quite long. Since each bin has data dependencies, each regular bin engine should be connected serially to process multiple regular bins. In this case, processing throughput increases but operating frequency decreases due to long critical path. This may reduce overall performance.

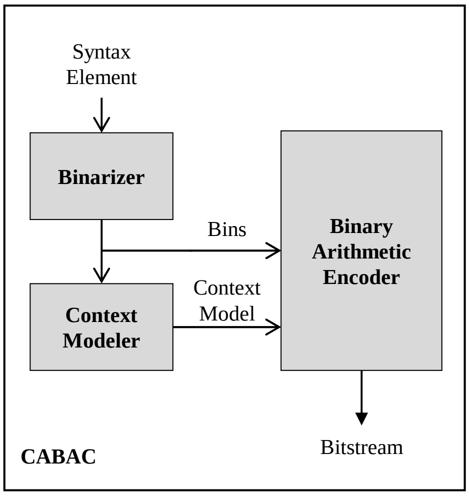

Figure 1. CABAC General Architecture

Fortunately, there is bypass mode in CABAC. In bypass mode, each bin has fixed probability. This means that each bin does not need to update context model. CABAC updates only *low* in bypass mode. So its computation is relatively simple with relatively short critical path. In addition, occurrence of bypass bins increases when image size is big or quantization parameter (QP) is small. [5] If video size is big or QP is small, the amount of information could extremely increase and it needs faster performance. So processing multiple bypass bins can be more efficient than processing multiple regular bins.

## 2. Analysis of performance

To determine the number of multiple bypass bin engines, thorough analysis was performed based on the simulation of target sequences. Processing throughput and critical path delay are calculated for different number of bypass bin engines. These results are compared, and the number of bypass bin engines is determined considering hardware size.

Processing throughput means how many bins are processed per cycle. If input pattern is constant, bins per cycle would be also constant. However, in CABAC, input pattern is always different. Some binstrings are encoded by only regular mode, and some binstrings are encoded by only bypass modes, and some binstrings are encoded by both modes. Therefore, if SE has many successive bypass bins, processing throughput would be high due to multiple bypass bin engines. On the other hands, if SE has many succesive regular bins, processing throughput would be low. Critical path delay means how long time is delayed in the encoder. It is a reciprocal of the operating frequency.

Figure 2 shows the processing throughput for different target sequences and different number of bypass bin engines. HEVC standard simulation model HM [5] was used in the simulation. Three target sequences and three encoder sequences were used. QP = 0 is used, since this condition maximizes the number of bins and minimizes the processing throughput. As shown in Firgure 2, processing throughput increases along with the number of bypass bin engines, but it becomes saturated when the number of bypass bin engines is 8 or more.

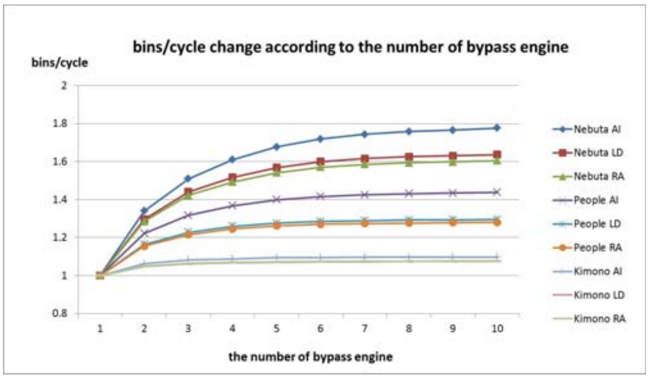

Figure 2. Performance throughput (bins /cycle)

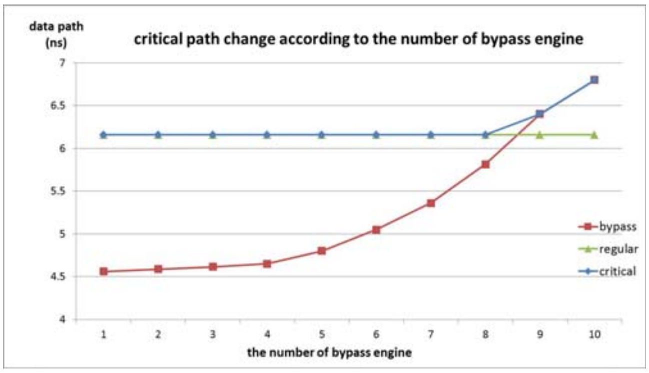

Figure 3. Critical Path Change

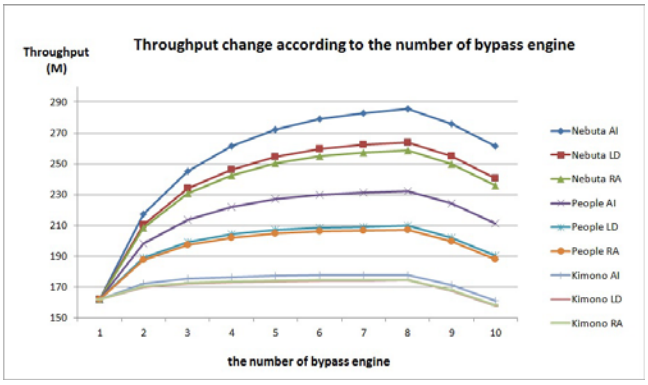

Figure 4. Throughput Change

Figure 3 shows the critical path delay for different number of bypass bin engines. Datapath of both regular bin engines and bypass bin engines are synthesized in 0.18 um fabrication technology to get the critical path delay. If 8 orless bypass bin engines are used, maximum frequency would not decrease becuase critical path is still regular engine. So performance can increase along with the number of bypass bin engines. However, bypass bin engines become critical path when 9 or more bypass bin engines are used. From this moment, maximum frequency decreases along with the number of bypass bin engines.

From Figures 2 and 3, processing throughput increases significantly and operating frequency remains unchanged when the number of bypass bin engines increases up to 4. From 5 to 8 bypass bin engines, processing throughput still increases but it does not improve much. Over 9 bypass bin engines, operating frequency significantly decreases, which decreases overall performance.

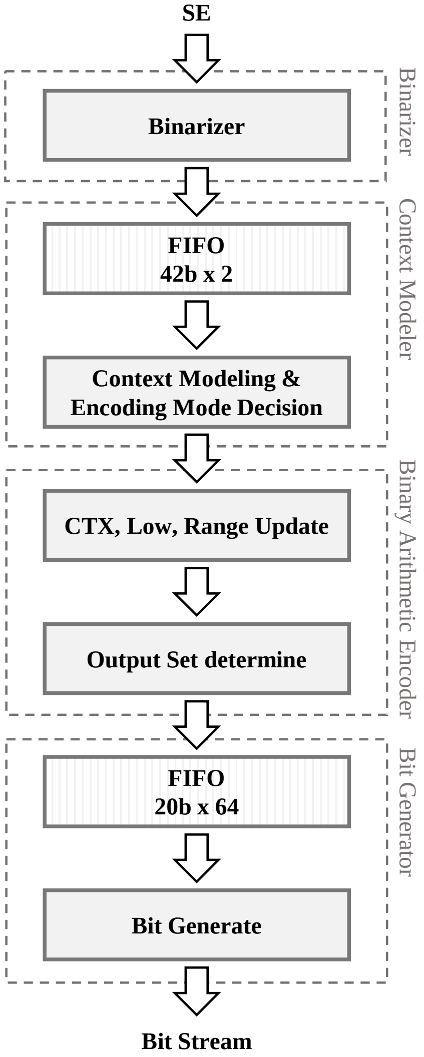

Figure 5. Proposed CABAC Top Architecture

As a result, we can get throughput Change like to Figure 4. if we use under 4 bypass, throughput / s could greatly increase, and from 4 to 8, throughput / s could increase in some sequences. Hardware size is very important in SoC implementation, since it is the main factor in the chip cost. It increases along with the number of bypass bin engines. However, these sequences need very high throughput because most of them has high resolution. So we designed 8 bypass engines CABAC to get highest performance for 4k resolution video.

## 3. Hardware Architecture

The Proposed CABAC encoder has 4 main modules, BI, CM, BAE, and bit generator (BG). In BAE, generating bitstream can be took from 0 to 7 cycle depends on regular mode renormalization loop. So as to avoid stall from this loop we separate BG from BAE. Instead, BAE calculates number of loop and variation (Output set). It is generated from each engine. So we can get maximum 8 output set. BG generates bitstream using this output set. This proposed architecture is shown in Figure 5. There are 2 FIFO memories to control data processing rate. The proposed BAE is shown in Figure 6. It has 1 regular engine, 1 terminate engine, and 8 bypass engine. Among each bypass engine there are data dependencies since each B engine updates low value every time and generates output set using updated low value. But it doesn’t need to calculate loop because renormalization occurs only one time in bypass mode. So its computation is not complicated. However, R engine updates not only low value but range value and context model (CTX). In addition, every updating process needs to consider loop condition and calculate the number of loop.  Since terminate bin occur one by one, we don’t need to process multiple terminate bin.

## 4. Simulation and Implementation

The implemented CABAC encoder has 8 bypass bin engines. It was designed in Verilog HDL under the CAD tool support of IDEC (IC Design Education Center). Based on HM [6], a standard reference HEVC software model, test vectors were generated, and the proposed CABAC encoder had passed all test vectors on FPGA. It was synthesized in 0.18 um fabrication technology. It has 69,577 gates including memory. Its maximum operating frequency is 162 MHz, and its maximum throughput is 284.8Mbins / s.

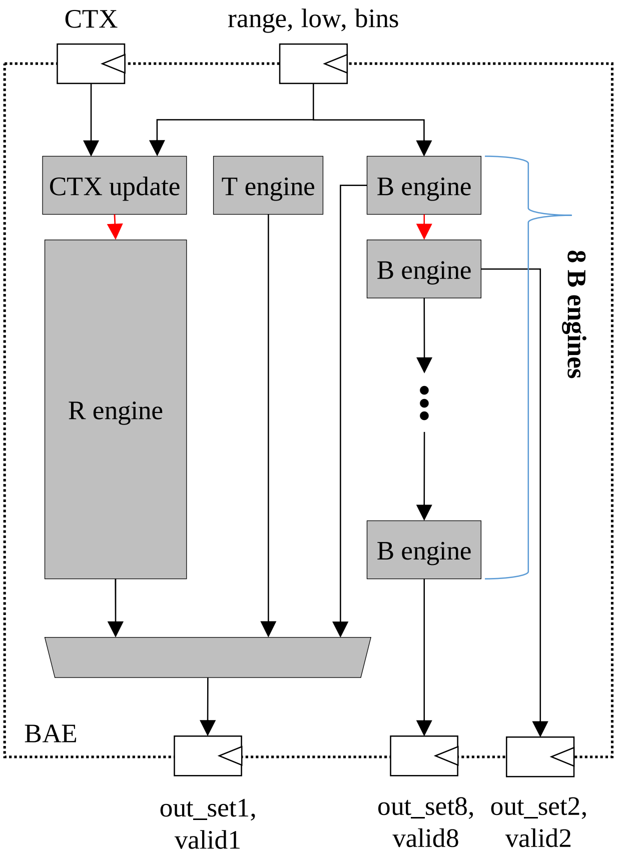

Figure 6. Proposed BAE Architecture

Table 1. Implementaion result

| | | |
|---|---|---|
| Technology (um) | | 0.18 |
| Maximum Operating Frequency (MHz) | | 162 |
| Maximum Processing Speed (Mbin/s) | | 284.8 |
| Gate Counts (gate) | Total | 69,577 |
| | Binarizer | 4,048 |
| | Context Modeler | 27,712 |
| | Binary Arithmetic Encoder | 37,780 |

Table 2. Simulation result

| sequence | configuration | QP | Throughput (Mbins / s) |
|---|---|---|---|
| Nebuta Festival (10 bit, 2560×1600 pixel,60 frame/s) | Intra Main | 0 | 284.8 |
| | | 7 | 239.7 |
| | | 22 | 201.4 |
| | Random Access | 0 | 258.2 |
| | | 7 | 221.6 |
| | | 22 | 185.1 |
| People On Street (8 bit, 2560×1600 pixel, 30 frame/s) | Intra Main | 0 | 236.8 |
| | | 7 | 204.4 |
| | | 22 | 187.8 |
| | Random Access | 0 | 206.8 |
| | | 7 | 190.2 |
| | | 22 | 183.1 |
| Basketball Drive (8 bit, 1920×1080 pixel, 50 frame/s) | Intra Main | 0 | 229.5 |
| | | 7 | 192.3 |
| | | 22 | 174.7 |
| | Random Access | 0 | 208.2 |
| | | 7 | 181.9 |
| | | 22 | 179.0 |

## Acknowledgements

This research was supported by Industrial Core Technology Development Program (10052009) and System IC Commercialization R&BD Program (10049498) funded by the Ministry of Trade, industry & Energy, Korea. It was also supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Korea (2014R1A1A2059625).

## References

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

[2] D. Kim, J. Moon, and S. Lee, “Hardware Implementation of HEVC CABAC Context Modeler”, Journal of IKEEE, vol. 19, no. 2, pp. 254-259, Jun. 2015.

[3] D. Pham, J. Moon, D. Kim, and S. Lee, “Hardware Implementation of HEVC CABAC Binary Arithmetic Encoder”, Journal of IKEEE, vol. 18, no. 4, pp. 630-635, Dec. 2014.

[4] D. Pham, J. Moon, D. Kim, and S. Lee, “Design of HEVC CABAC Encoderwith Parallel Processing of Bypass Bins”, Journal of IKEEE, vol. 19, no. 4, pp. 583-589, Dec. 2015.

[5] Joel Sole, Rajan Joshi, Nguyen Nguyen, Tianying Ji, Marta Karczewicz, Gordon Clare, Felix Henry, and Alberto Duenas, “Transform Coefficient Coding in HEVC”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1765-1777, Dec. 2012.

[6] HEVC software repository HM-11 reference model at https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/branches/HM-11.0-dev/