High Speed Decoding of Context-based Adaptive Binary Arithmetic Codes Using Most Probable Symbol Prediction

Chung-Hyo Kim
Power Generation Lab
Korea Electric Power Research Institute
Daejeon, Korea
Email : ch2kim@kepri.re.kr

In-Cheol Park
Electrical Engineering and Computer Science
Korea Advanced Institute of Science and Technology
Daejeon, Korea
Email : icpark@ee.kaist.ac.kr

***Abstract*—Context-based Adaptive Binary Arithmetic Coding (CABAC) is the major entropy-coding algorithm employed in H.264/AVC. Although the performance gain of H.264/AVC is mostly resulted from CABAC, it is difficult to achieve a fast decoder because the decoding algorithm is basically sequential. In this paper, a prediction scheme is proposed to enhance overall decoding performance by decoding two binary symbols at a time. A CABAC decoder based on the proposed prediction scheme improves the decoding performance by 24% compared to conventional decoders.**

In this paper, we propose a parallel CABAC decoding method that can decode two binary symbols at a time to achieve a high-speed decoder meeting the requirement of the H.264/AVC standard. In the proposed decoding method, the first binary symbol is decoded as the conventional scheme, while the second is decoded with predicting that the first symbol is the most probable one. We can decode two symbols simultaneously if the prediction is valid. Experimental results show that the proposed prediction scheme improves decoding performance by 24% compared to conventional decoding methods.

I. INTRODUCTION

The newest international video coding standard H.264/AVC developed by the joint video team of the MPEG and ITU can produce a perceptually equivalent quality video at about half the bit-rate compared to MPEG-2. The performance gain is mainly resulted from context-based adaptive binary arithmetic coding (CABAC) employed in H.264/AVC main profile [1]. The CABAC, a binary arithmetic code [2] associated with the context modeling technique, was reported in [3] that it can save up to 32% bit-rate compared to other compression methods such as Huffman [4] and Exp-Golomb codes [5] if appropriate context models are provided. Therefore, H.264/AVC is adopted in a diverse range of multimedia applications, including HD-DVDs, HDTV broadcasting, and internet video streaming.

Although more than 90 percents of the H.264 main profile stream is encoded using the CABAC, its decoding algorithm is basically sequential and needs large computation to calculate range, offset and context variables, making it difficult to achieve high decoding performance [6]. The CABAC decoding complexity required to process high definition images in real time is about 3 giga-operations per second. Although this computing complexity is still less than the block processing complexity, the CABAC decoding becomes a major bottleneck in real time processing due to its sequential nature. On the other hand, the block processing can be easily enhanced by applying parallel and pipeline techniques.

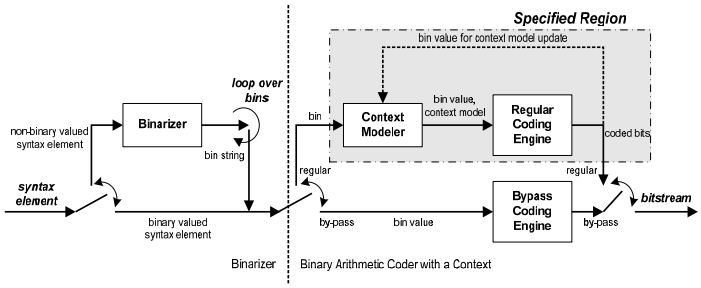

Fig. 1. The CABAC encoding procedure

II. CABAC ENCODING AND DECODING

This section describes the encoding and decoding procedure of CABAC briefly, as it is essential to understand the proposed prediction-based decoding scheme.

A. CABAC Encoding

Fig. 1 shows the encoding procedure of CABAC. A sequence of syntax elements to be encoded is first converted to a sequence of codewords, each of which is a binary string consisting of binary symbols called bins, as CABAC deals with only binary symbols. The symbol with the higher probability is called the most probable symbol (MPS) and the other is the least probable symbol (LPS). Syntax elements that are already expressed in binary strings can skip this binarization step. Before applying binary arithmetic coding, a specific context model containing the LPS

This work was supported in part by University IT Research Center Project, and by Korea Science Engineering Foundation through the MICROS center.

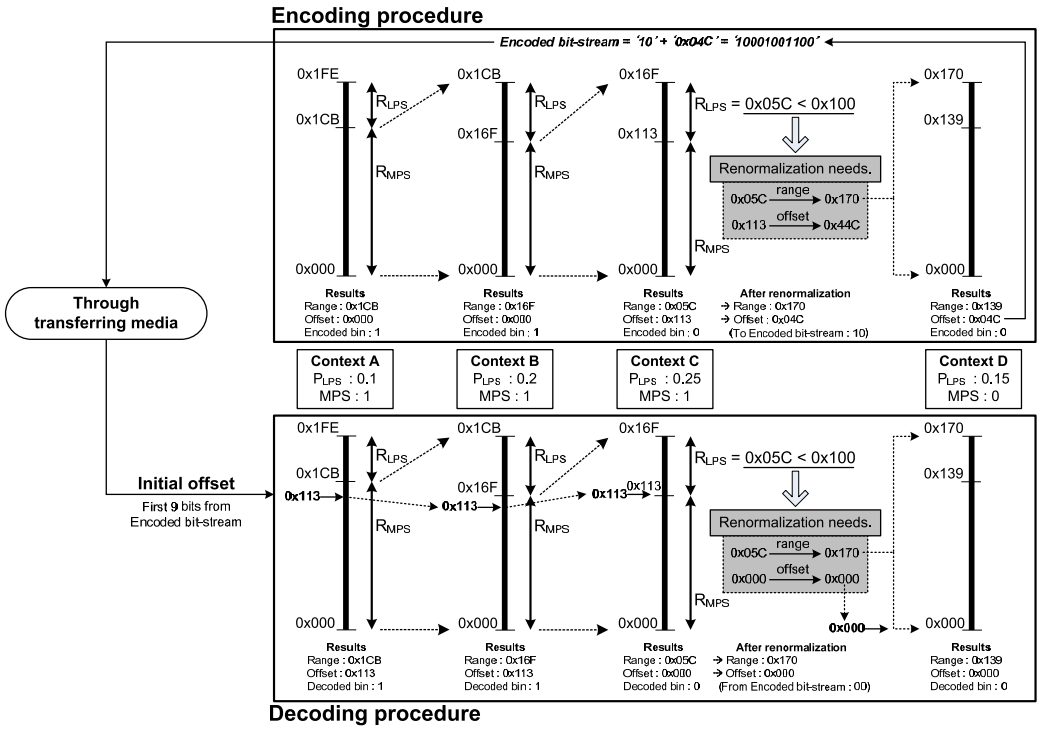

Fig. 2. The encoding procedure

probability and the MPS value is assigned to each bin. Each bin associated with a context model is sequentially encoded to produce an encoded bit-stream.

Since the CABAC is effective when the bins to be encoded are biased to certain values, a sequence of bins can bypass the encoding when the sequence is not biased, that is, when the 0's and 1's in the pattern are evenly distributed. The CABAC is based on a binary arithmetic coder that allows substantial complexity reduction with high compression efficiency. More specifically, the binary arithmetic coder is related to the Q-coder family [7]. The context model is not fixed but adaptively updated for the next encoding. If the MPS is encoded, the LPS probability of the context decreases, otherwise, it increases. To encode a bin, the binary arithmetic coder (BAC) needs the corresponding range and context model. The range is to indicate an interval. Starting from the initial range, 0x1FE, it is narrowed after each bin is encoded. The range is divided into two sub-ranges, R_{MPS} and R_{LPS}, where R_{LPS} is calculated by multiplying the range and the LPS probability specified in the context model, and R_{MPS} is computed by subtracting R_{LPS} from the range. In fact, a two-dimensional table that can be indexed with the LPS probability quantized to 6 bits and two most significant bits of the range is used to replace the multiplication. Since the range becomes narrow as the decoding progresses, more bits are needed to represent it. However, the range and offset are required to be limited to 9 bit in CABAC. To cope with this situation, the range is renormalized to make it equal to or greater than 0x100. If the range is less than 0x100, it is shifted left until the range is in [0x100, 0x1FE]. An encoding example is shown in the upper part of Fig. 2, which corresponds to the shaded part of Fig. 1. The bins to be encoded are '1100' and the encoded bit-stream, 0x44C, is transferred through storage or wireless media.

B. CABAC Decoding

The decoding is similar to the encoding. Given an encoded bit-stream, a CABAC decoder repeatedly decodes bins. The CABAC decoder has a merging unit to check whether the sequence of decoded bins matches with a meaningful codeword. Except the merging unit, the encoding and decoding procedures are almost the same. To decode a bin, the binary arithmetic decoder (BAD) needs the corresponding range, offset and context model. The offset is criterion for deciding decoded bin, and initialized by taking the first 9 bits from the encoded bit-stream. A decoding example is shown in Fig. 2, where the initial offset is set to 0x113. Note that a different context model can be used for each bin decoding. If the offset is less than R_{MPS}, the bin is the MPS and the range to be used for the next decoding is set to R_{MPS}. Otherwise, the bin is determined to the LPS, the inversion of the MPS value contained in the associated context model, and the next range is set to R_{LPS}. As in the encoding procedure, renormalization is required to limit the range and offset to 9 bits. The offset is renormalized by appending lower *n* bits from the encoded bit-stream, where *n* is the shift amount.

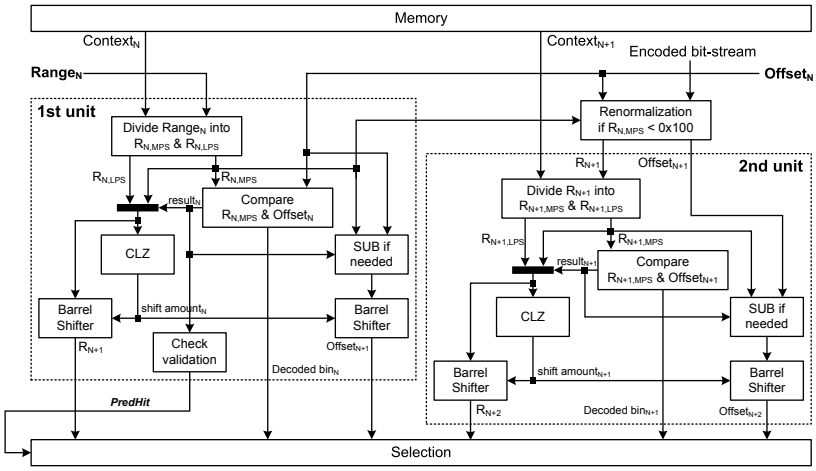

Fig. 3. The proposed CABAC decoder

III. PROPOSED PREDICTION SCHEME

This section describes the proposed prediction-based CABAC decoding scheme, which is based on the analysis of how and what values are updated after a single bin is decoded

A. Analysis of variable changes

If the decoded bin is the MPS and renormalization does not occur, the next range is set to R_{MPS}, and the same offset is used for the next bin decoding. The value of R_{MPS} is in [0x080, 0x1FE], because the MPS probability is always greater than or equal to 0.5. Therefore, just one shift is sufficient for renormalization when the most significant bit of R_{MPS} is zero. Otherwise no addition operation is needed for renormalization.

In case of LPS decoding, the next range is set to R_{LPS}. As the LPS probability is not greater than 0.5, the value of R_{LPS} is in [0x000, 0x0FF], which means that the renormalization occurs always and requires multiple shifts. For the next decoding, it is needed to count the leading zeros and shift the R_{LPS} to the left with that amount. For example, the third decoding in Fig. 2 requires 2 left shifts to renormalize the range and offset. In addition, the offset should be adjusted by subtracting the value of R_{MPS} before the renormalization. TABLE 1 summarizes how variables are updated according to the decoded bin.

TABLE 1
Variable update after one bit decoding

Case		MPS decoding	LPS decoding
No renormalization	Frequency	Frequent	None
	Range	R_{MPS}	-
	Offset	No change	-
Renormalization	Frequency	Rare	Always
	Shift amount *n*	1	Arbitrary
	Range	$R_{MPS} << 1$	$R_{LPS} << n$
	Offset	Offset << 1	$(Offset - R_{MPS}) << n$

If the decoded bin is the LPS, it takes a lot of time to calculate the next range and offset required for the next bin decoding. The range always has to be modified and the offset is also subtracted. To find the shift amount *n*, the decoder has to be equipped with a unit that counts leading zeros (CLZ). In addition to that, it needs an 8-bit barrel shifter to finalize renormalization. If the decoded bin is the MPS, however, the next range and offset can be calculated with simple operations even when the renormalization is involved. After calculating R_{MPS}, at most one shift is enough for renormalization.

Since CABAC is a binary arithmetic coding which deals with binary symbols, it seems that the next bin decoding can start earlier by assuming the current bin has a specific value, the LPS or MPS. Although the variables required for the next bin decoding are updated and dependent on the current bin decoding result, we can get the variables in advance by predicting the result of the current bin decoding. If the LPS is decoded in the first decoding, it takes a long time to calculate the range and offset. In this case, there is no difference from the traditional decoding scheme that decodes bins in sequel. However, if the MPS is decoded, the variables for next bin decoding can be calculated with simple operations.

Based on the analysis, we propose a prediction scheme to decode two bins at a time. The first binary symbol is decoded as the conventional scheme, while the second is decoded with predicting that the first symbol is the MPS.

B. Patterns of neighboring context

There is a problem to be solved for the proposed parallel decoding. To start the second predicted decoding, we should know which context model is to be used for the second bin decoding. Since the context model is selected out of 399 ones by referring to the type of the syntax element or the previous bin decoding, it is difficult to determine the context model without knowing the result of the first bin decoding.

TABLE 2
Simulations for two benchmark files

File	Case	No. of bins	Prediction hit (%)	bins/cycle
Car.yuv	Predictable	14499682	69.07	0.56
	Unpredictable	13745309	-	0.33
	Total	28244991	-	**0.41**
Cheer.yuv	Predictable	20441764	65.42	0.55
	Unpredictable	20567685	-	0.33
	Total	41009449	-	**0.41**

In addition, a context model used for a bin decoding is updated after the bin is decoded, which means that the updated context model should be used for the second bin decoding if the same context model is applied to the first and second bins. Analyzing the pattern of context models applied to two sequential bins contained in a syntax element, we found that the two context models are different in most cases and the second context model is usually next to the first context model. Based on the observation, the second bin is decoded with predicting that the first bin is the MPS and the next context model is used.

C. The proposed CABAC decoder

The proposed CABAC decoder shown in Fig. 3 includes two BADs and reads two sequential context models at a time. The first BAD is the same as the conventional decoder except additional two output signals, $R_{N,MPS}$ and PredHit, where $R_{N,MPS}$ is the range to be transferred to the second BAD and PredHit is to indicate whether the decoding result of the first BAD is the MPS and the next context model is valid for the second bin decoding. Before starting the second bin decoding, the second BAD performs renormalization if $R_{N,MPS}$ is less than 0x100. In this case, at most one shift is enough. If PredHit is asserted, the decoding result of the second unit is valid. Otherwise, it is discarded. Since there are little changes in the range and offset when the MPS is decoded in the first unit, the second unit can start after a little delay needed to calculate $R_{N,MPS}$.

IV. EXPERIMENTAL RESULTS

The proposed decoder was described in Verilog HDL and synthesized in 0.18um CMOS technology. The bin decoding is processed in three cycles. Two context models are read in the first cycle, bins are decoded in the second cycle, and the decoded bins are merged in the third cycle. The critical part of a conventional decoder is the BAD unit that takes 3.3ns. As the second BAD needs some additional delay to calculate $R_{N,MPS}$ and select valid results, its delay increases to 4.5ns in the proposed decoder. To reduce the additional delay in the second cycle, the calculation of $R_{N,MPS}$ and the selection of valid results are retimed to the first cycle and the third cycle as shown in Fig. 4. With the retiming, the proposed decoder can work at the same clock frequency as that of the conventional decoder.

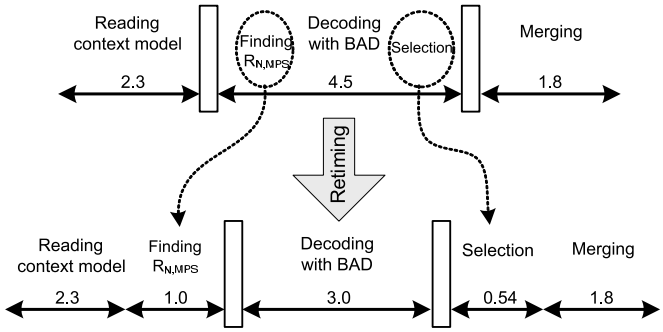

Fig. 4. Retiming to reduce critical path delay (unit:ns)

TABLE 2 shows the simulation results for two benchmark image files. The case that has at least two bins remaining to be decoded in a syntax element is called a predictable case, and the other is an unpredictable case. The prediction is hit only when the second predicted decoding is valid, that is, when the first BAD decodes the MPS and the next context model is valid for the second decoding. If the prediction is hit, we can decode two bins at a time and thus save three cycles. The prediction accuracy is 67% on the average if predictable cases are considered. With the proposed predicted decoding, we can decode 24% more bins compared to the conventional serial decoding.

V. CONCLUSION

We have presented a prediction-based CABAC decoding scheme to improve the performance by decoding two bins simultaneously. In the proposed scheme, the second bin is decoded with predicting that the first bin is the MPS and the context model is next to that used for the first bin. Experimental results show that the proposed prediction scheme can improve the overall decoding performance by 24% compared to conventional decoders.

REFERENCES

[1] D. Marpe, H. Schwartz, and T. Wiegand. "Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC video compression standard," IEEE Trans. on CSVT, vol. 13, pp. 620-636, July 2003.

[2] I. H. Witten, R. M. Neal, and J. G. Cleary. "Arithmetic coding for data compression," Communications of the ACM, vol. 30, pp. 520–540, June 1987.

[3] D. Marpe, G. Blattermann, G. Heising, and T. Wiegand. "Video compression using context-based arithmetic coding," ICIP 2001, vol. 3, pp. 558-561, Oct. 2001.

[4] R. D. Hoffman, "A method for the construction of minimum redundancy codes," Proc. IRE, vol. 40, pp. 1089-1101, Sept. 1952.

[5] J. Teuhola, "A compression method for clustered bit-vectors," Inform. Proceedings Lett., vol. 7, pp. 308-311, Oct. 1978.

[6] H. Eeckhaut, H. Devos, B. Schrauwen, M. Christiaens, and D. Stoobandt, " A hard-ware-friendly wavelet entropy codec for scalable video," Proceedings of Design, Automation and Test in Europe, vol. 3, pp. 14-19, March 2005.

[7] W. B. Pennebaker, J. L. Mitchel, G. G. Langdon jr., and R. B. Arps. "An overview of the basic principles of the Q-coder adaptive binaey arithmetic coder", IBM Journal of Research and Development, vol. 32(6), pp. 717-726, November 1988.