
High speed CABAC codecs for DSP- and FPGA-based platforms

Damian Karwowski and Marek Domański

Poznan University of Technology,
Chair of Multimedia Telecommunications and Microelectronics
dkarwow@multimedia.edu.pl
domanski@et.put.poznan.pl

Summary. In this paper, two original structures of Context-based Adaptive Binary Arithmetic Codec (CABAC) are proposed for real-time video compression systems. Two proposed CABAC codecs were implemented and optimized for DSP-based platform and FPGA-based platform respectively. Analysis of implementations of proposed CABAC codecs was done. Both the software and hardware version of CABAC codec was tested in terms of entropy codec throughput. Number of clock cycles needed to process a binary symbol was measured for software version of CABAC as well as hardware version of CABAC.

1 Introduction

In order to improve compression ratio of video encoders, entropy coding is commonly used. The state-of-the-art entropy coding technique used in video compression is the Context-based Adaptive Binary Arithmetic Coding (CABAC) [1, 3] which became a part of new worldwide Advanced Video Coding (AVC) standard (ISO MPEG-4 AVC and ITU-T Rec. H.264) [1].

In comparison to other entropy coders used in video compression, CABAC algorithm is distinguished due to significantly higher coding efficiency. Nevertheless, computational complexity of CABAC codec is also significantly higher. Important research problem is to develop computationally efficient structure of CABAC codec that will be able to process high quality video in real-time.

This paper presents two original structures of CABAC codec that were optimized for DSP-based platforms and FPGA-based platforms respectively. Two proposed structures of CABAC codec were implemented from scratch and tested. Analysis of performed implementations was done. The throughput of software version as well as hardware version of CABAC codec was measured.

2 CABAC entropy coding algorithm

As stated above, CABAC algorithm is the most sophisticated entropy coding technique ever applied in compression of digital video. It uses advanced mechanisms of data statistics estimation and efficient arithmetic coding. Three elementary functional blocks can be distinguished in CABAC. These are: the binarizer of input symbols, the context modeler and binary arithmetic codec (see Figure 1).

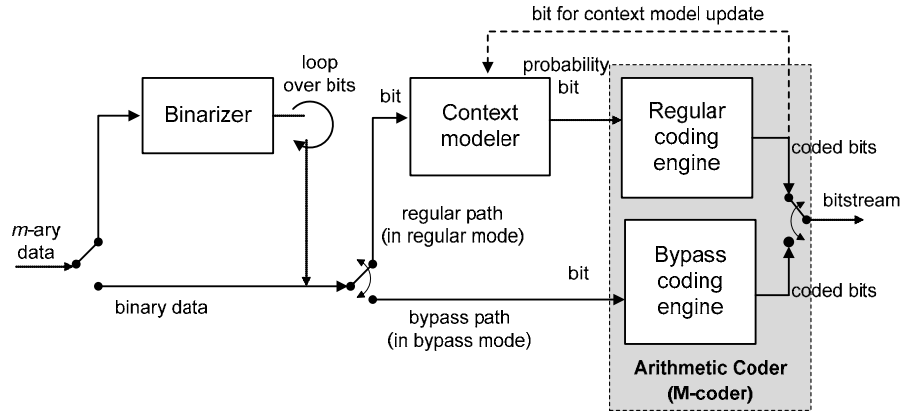


Fig. 1. CABAC encoder block diagram

In order to increase throughput of a CABAC codec, fast core of binary arithmetic codec (the so-called M-codec) has been used [8, 9]. In CABAC, due to application of binary arithmetic coding, all non-binary valued syntax elements have to be represented with a string of binary symbols. This mapping process of symbols is realized by a binarizer. For the reason that the binarization greatly influences the size of bitstream at the output of entropy codec as well as its computational complexity, five different binarization schemes are used in CABAC [3]. The binarization works similar as variable-length coding (e.g. Huffman coding) but in contrast to it the inter-symbols redundancy is extra reduced with efficient arithmetic coding.

Arithmetic codec processes the binary symbols with respect to the conditional probabilities of their occurrence in video data stream. The actual arithmetic coding has already been thoroughly studied. There exists powerful cores of arithmetic codecs (in terms of coding efficiency), so the actual arithmetic coding is currently not a topic of intensive research works. Compression capabilities of adaptive arithmetic encoders strongly depend on mechanisms of data statistics estimation used in the context modeler that calculates probabilities of symbols. The mechanisms of adaptation to the current signal statistics that are used in CABAC belong to the most advanced ones. In order to adapt

better to changing statistics of coded data as many as 399 separate probability models are used for all coded syntax elements (this is only for the case of a DCT-like transform calculated in 4x4 blocks). Each probability model exploits pre-defined probability distribution of data. A given syntax element uses some sub-set of all possible probability models. The individual probability model is chosen with exploiting the statistics of this element in neighbouring blocks within an image. It allows the adaptation algorithm to adapt rapidly to the current statistics of two-dimensional signal. Thus, the chosen probability model is used to estimate the probability of symbol that appeared in a given context. This probability is finally used by arithmetic codec core.

3 The research problem

As stated above mechanisms of data statistics modeling used in CABAC exploits adaptive models of source data that belongs to the most efficient ones ever applied in digital video coders. This feature results in considerably higher compression efficiency than in any other entropy coder commonly used in video compression [3].

However, high coding efficiency of CABAC algorithm has been achieved by a significant increase of computational complexity of entropy encoding and entropy decoding. Real-time high quality video compression with CABAC algorithm is a great challenge even for today's powerful processing units.

Regarding to this, an important research problem is to propose the optimized architecture of adaptive entropy coders that will enable real-time processing of high quality video. Two optimized architectures of CABAC are presented: software version of CABAC codec and hardware version of CABAC.

4 Software version of CABAC codec

Application of sophisticated mechanisms of data statistics modeling in CABAC makes its implementation far more difficult and far more time-consuming relative to other entropy codecs used in video compression. The original software version of CABAC codec (encoder and decoder) has been proposed and implemented in C programming language. All components of new CABAC codec have been algorithmically highly optimized towards speed. The difficulty and the outlay of work of implementing CABAC encoder and CABAC decoder are comparable. The original implementation of the optimized CABAC codec contains approximately 5200 lines of program code in C. The core of the binary arithmetic codec contains only 380 lines of program code. Thus, from the point of view of implementation of CABAC codec the core of arithmetic codec makes only about 7% of the whole CABAC implementation. It means that implementations of other parts of CABAC codec (binarization, context model selection and probability estimation and update) make over 90% of the

whole CABAC implementation. These figures are similar to those obtained for other commonly known implementations of CABAC codec. For comparison, in the implementation of CABAC from x264 video codec [10] the core of arithmetic codec makes approximately 9% of the whole CABAC and in the implementation of CABAC in JM 10.2 reference software [2] the core of arithmetic codec makes about 12% of the whole CABAC (see Figure 2).

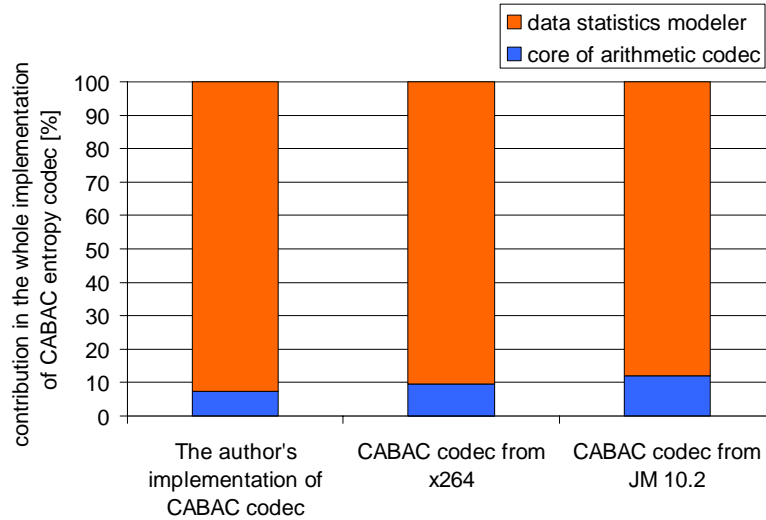


Fig. 2. Contribution of arithmetic codec core in three implementations of CABAC codec

Thus, the block of data statistics modeling makes an essential part of CABAC codec which determines its complexity and compression ratio to a large extent. Complex mechanisms of data statistics gathering together with binary arithmetic coding lead to a considerably smaller amount of computations required in processing symbols by CABAC. Presented software version of CABAC codec was successfully activated in original fast AVC video decoder dedicated to signal processor platforms. Measurements on complexity of original software CABAC working within fast AVC decoder revealed that high-performance digital signal processor TMS320DM642 [11] (with frequency of 600 MHz) is able to entropy decode only bitstreams with bitrates below ten megabits per second. Decoding of a binary symbol with CABAC absorbs about 75 cycles of TMS320DM642 processor. Experimental results showed that different processor power is needed for data statistics modeling and binary arithmetic coding. In original implementation of CABAC actual binary arithmetic coding makes about 40% of total CABAC decoding time of a binary symbol. Other parts of CABAC decoder (de-binarization and data statistics estimation) make about 60% of total CABAC decoding time (see Figure 3).

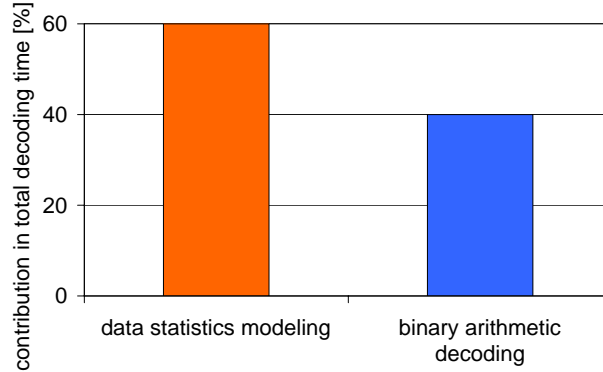


Fig. 3. Contribution of data statistics modeling and binary arithmetic decoding in total decoding time of a binary symbol

Thus, data statistics modeling is considerably more time-consuming than binary arithmetic decoding in CABAC. It leads to a fact that CABAC coding is a processor-intensive task that demands extremely high-performance digital processors for real-time coding of high quality video. Moreover, the serial nature of CABAC makes it impossible to efficiently exploit today's high power digital media processors with multi-core technology. Therefore, intensive research is being conducted these days to propose the modified CABAC algorithm that will be able to do computations in parallel [12, 13, 14]. These works are being done in the context of prospective new video coding standard H.265 [15]. Nevertheless, it is commonly known that Field Programmable Gate Arrays (FPGA) platforms are characterized by considerably higher processing capabilities in comparison to digital signal processors. Therefore, power demanding advanced entropy coders can be more efficiently realized in hardware. It was the motivation for authors to propose the hardware version of CABAC codec.

5 Hardware version of CABAC decoder

5.1 Implementation of CABAC entropy decoder

A hardware version of CABAC decoder was designed and implemented. In this implementation, CABAC decoder has been clearly divided into three main functional blocks: a block of arithmetic decoder core, a block of de-binarization and control of syntax elements decoding and a block of local context management. The task of de-binarization and control of syntax elements decoding is realized with two functional blocks: a syntax elements decoding and a transform coefficients decoding. The transform coefficients decoding block realizes de-binarization and decoding of block of transform coefficients. The syntax

elements decoding block controls the process of de-binarization and decoding of all remaining syntax elements. Owing to the application of several different binarization schemes for syntax elements in CABAC, ROM memory which contains the methods of decoding and de-binarization of individual syntax elements has been used. In order to decode a binary symbol, syntax elements decoding and transform coefficients decoding modules strobe the arithmetic decoder core module that realizes arithmetic decoding of symbols, taking into account the proper probability model saved in context models block. The number of probability model is calculated by managing local context module based on the values of symbols in neighboring blocks. The core of arithmetic decoder decodes encoded bitstream from input buffer block that is filled with bitstream of encoded data. The block diagram of original hardware CABAC decoder has been presented in Figure 4.

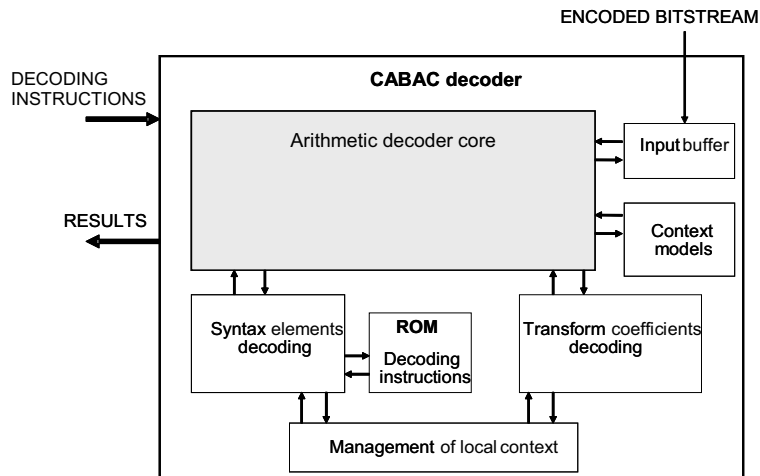


Fig. 4. General block diagram of original hardware version of CABAC decoder

CABAC algorithm in a great deal exploits dependencies between symbols. For that reason, both CABAC encoding and CABAC decoding is a sequential process. It is very difficult to do computations parallelly in CABAC. Nevertheless, there are some possibilities to accelerate CABAC coding in hardware realization. In general terms, CABAC decoding of a binary symbol can be divided into the following tasks:

- Calculating of the number of probability model;
- Arithmetic decoding of a binary symbol with respect to the probability model;
- Renormalization of registers of arithmetic decoder core;
- Updating the probability model with respect to the value of decoded binary symbol.

In original implementation of CABAC decoder these computations are performed with exploiting parallelism and tasks pipelining. When arithmetic decoder decodes the current symbol, computations for register renormalization and the probability model updating can be done in parallel. Calculation of the number of probability model for the successive binary symbol can also be lunched at the same time. After calculating the number of probability model for the successive symbol, it has to be checked if the process of renormalization of registers of arithmetic decoder core has already been finished. If it yes, the core of arithmetic decoder can do the decoding of the successive symbol. In this way, the throughput of CABAC decoder has been significantly increased.

5.2 Features of original hardware version of CABAC decoder

The original implementation of hardware CABAC decoder contains about 5500 lines of program code written in Verilog [4] hardware description language (HDL). The project has been synthesized on Virtex 5 FPGA platform [5] with ISE 9.2i software [6]. The maximum clock frequency of CABAC decoder is 192 MHz and it utilizes about 1600 Virtex 5 slices. It is commonly known that approximately three times higher performance can be achieved when realizing the design as an application-specific integrated circuit (ASIC) [7]. According to that, the original CABAC decoder realized as an ASIC can work with maximum frequency of about 600 MHz when using the same process technology as FPGA platform.

The tests on the performance of hardware CABAC decoder with a set of a hundred thousands binary symbols have been done. Experimental results revealed that the original hardware CABAC decoder decodes a binary symbol in 7.5 clock cycles in average. For comparison, high-performance digital media processor TMS320DM642 needs about 75 clock cycles to decode a binary symbol for original software implementation of CABAC decoder. The throughput of hardware version of CABAC is approximately ten times bigger than the computational performance of software version of CABAC. Original hardware CABAC decoder realized as an ASIC circuit (with frequency clock of 600 MHz) is able to process video bitstreams with bitrates of up to 80 megabits per second in real-time.

6 Advanced adaptive entropy coding - conclusions

Experimental results presented in the paper proved that contemporary adaptive arithmetic coders used in video compression are characterized by high computational complexity. Research was done in the context of CABAC coder which is the state-of-the-art entropy coding technology. Two original versions of CABAC codec were proposed: software and hardware version of CABAC. The proposed implementations of CABAC were highly optimized towards speed and were tested in terms of the codec throughput.

Proposed software version of CABAC activated on TMS320DM642 processor is able to process video bitstreams with bitrates below 10 megabits per second in real-time. In this implementation data statistics estimation makes about 60% of total entropy coding time.

The throughput of CABAC codec can be significantly increased when realizing on hardware platforms. This is an important conclusion of the paper. Proposed hardware version of CABAC decoder was successfully synthesized on Virtex 5 FPGA platform at maximum clock frequency of 192 MHz. Experimental results revealed, that hardware version of CABAC is able to process a binary symbol in ten times smaller number of clock cycles relative to its software version.

References

1. ISO/IEC 14496-10, "Generic Coding of Audio-Visual Objects, Part 10: Advanced Video Coding", March 2006.
2. H.264/AVC software coordination site - <http://bs.hhi.de/~suehring/tml>.
3. D. Marpe, 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, Vol. 13, No. 7, pp. 620-636, July 2003.
4. IEEE Standard Hardware Description Language Based on the Verilog Hardware Description Language, IEEE Std 1364-1995, IEEE, 1995.
5. Virtex-5 FPGA Family Datasheet. Xilinx, Inc., San Jose, CA, 2007, <http://www.xilinx.com/>.
6. Xilinx ISE 9.2i software manuals, <http://www.xilinx.com/>.
7. I. Kuon, and J. Rose, "Measuring the Gap Between FPGAs and ASICs". IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., Vol. 26, No. 2, p. 203, February 2007.
8. D. Marpe, and T. Wiegand, "A highly Efficient Multiplication-Free Binary Arithmetic Coder and Its Application in Video Coding", IEEE International Conference on Image Processing (ICIP'03), Barcelona, Spain, September 2003.
9. D. Marpe, G. Marten, and H. L. Cycon, "A Fast Renormalization Technique for H.264/MPEG4-AVC Arithmetic Coding", 51st Internationales Wissenschaftliches Kolloquium Technische Universität Ilmenau, September 2006.
10. x264 Project. <http://www.videolan.org/developers/x264.html>.
11. Texas Instrument, TMS320DM642 Video/Imaging Fixed-Point Digital Signal Processor. www.ti.com, July, 2002.
12. V. Sze and M. Budagavi, "Parallel CABAC", ITU-T SGI 6/Q.6 Doc. COM16-C334, Geneva, Switzerland, April 2008.
13. J. Zhao, A. Segall, ITU-T VCEG SG 16 Q.6, Doc. VCEG-AI32, "New Results using Entropy Slices for Parallel Decoding", 35rd VCEG Meeting, Berlin, Germany, 16-18 July 2008.
14. X. Guo, Y.-W. Huang, S. Lei, ITU-T VCEG SG 16 Q.6, Doc. VCEG-AK25, "Ordered Entropy Slices for Parallel CABAC", 37th VCEG Meeting, Yokohama, Japan, 15-18 April 2009.
15. ITU-T VCEG SG 16 Q.6, Doc. VCEG-AG01, 33rd VCEG Meeting Report, Shenzhen, China, 20 October 2007.