IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

# STUDY OF CAVLC DECODER IN H.264/AVC VIDEO DECODING

**Saieel Vaigankar**
*Department of Electronics and Telecommunication*
*Goa College of Engineering, Farmagudi, Ponda, Goa, India*

**Abstract:** Video decompression (VC) is crucial in real-time High Definition (HD) applications. The earlier iteration of the different VC algorithms that are currently available was the H.264/AVC standard. It can handle HD images up to 60 frames per second, unlike the earlier AVC (Advanced Video Codec) standard. The principles of inter- and intra-prediction and CAVLC entropy coding (Context-Adaptive Variable Length Coding) underpin H.264/AVC. In contemporary video decompression systems, especially in standards like H.264/FFMPEG AVC, the CAVLC Decoder is essential. One major obstacle to attaining quick and effective video decoding is the computational complexity of CAVLC decoding. To speed up the CAVLC decoding process and advance video decompression technology, this study proposes to design a hardware IP core for the AVC video-coding standard. The proposed method uses the FFMPEG tool to generate bitstream, which is to be used in Verilog code for the CAVLC decoder.

Index Terms—CAVLC, AVC/H.264 standard, verilog, video decompress

## I. INTRODUCTION

Video compression methods are essential in the constantly changing world of digital media. The growing demand for high-quality video footage makes efficient video compression (VC) crucial for contemporary multimedia applications. One of the main components of the H.264/MPEG-4 AVC video compression standard is context-adaptive variable length coding or CAVLC. By lowering the size of video files and the bandwidth required for their transmission, efficient video compression (VC) is essential to addressing this demand. This bandwidth decrease makes it possible to broadcast video more quickly and consistently, even over slow internet connections. Additionally, it lessens the storage needed for video files, which is crucial for mobile devices with constrained storage.

Acknowledging these difficulties, the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) formed the Joint Video Team (JVT), which set out to create a new video compression standard that would rectify the flaws in the previous ones. Their work culminated in the 2003 introduction of the H.264/MPEG-4 AVC standard, which made H.264 available in 2003 and an enhanced version in 2013. It uses sophisticated compression techniques, such as 60 frames per second entropy coding (CAVLC). The ever-increasing need for high-quality video content and the inherent constraints of current compression techniques drove the development of AVC video coding. Particularly in situations where bandwidth was limited, traditional video coding techniques like MPEG-2 found it challenging to provide adequate compression efficiency while preserving a visually acceptable level of quality. Compared to HEVC, AVC has a more developed ecosystem, is more widely used, has lower computational complexity, is better compatible with older devices, has cheaper license fees, strong streaming performance, and an acceptable compression efficiency. For numerous applications, AVC is the recommended option because of these factors.

This entropy coding technique is one of two that the H.264/AVC standard offers; the other is called CAVLC (Context Adaptive Variable Length Coding). Both Huffman coding and binary arithmetic coding are employed as entropy coding techniques in the context of Advanced Video Coding (AVC), and they are essential for effectively encoding transform coefficients and other syntactic elements. When comparing the bit rates of videos encoded using AVC (H.264) and HEVC (H.265), several parameters are considered. These include the video's content, the encoding settings, and the desired compression efficiency. But generally speaking, HEVC has superior compression efficiency than AVC, which means it can achieve higher visual quality at the same bit rate or equal visual quality at lower bit rates. The compression efficiency of CAVLC is 10% to 40% higher than that of CABAC. Similar to CABAC, 30% to 50% compression efficiency. Compared to CAVLC, which has less complexity overall, CABAC requires more computational complexity.

When decoding data from the encoded bitstream of video frames, the context-adaptive variable length coding technique, or CAVLC decoder, modifies its coding scheme accordingly. The video frame's neighbouring pixels are used to calculate the probability distribution of the characters that are being encoded. Symbols with higher probability have shorter codes based on context information, while symbols with lower probabilities have more extended codes [9].

To our knowledge, this is the first publication on the CAVLC Decoder that implements it using low-level synthesis. Additionally, we use FFMPEG to extract the bitstream, which is then used as the binary input for upscaling Verilog code to maintain the same video quality without experiencing video loss. In the future, we'll use the Verilog code to accomplish these projects on the FPGA kit.

The paper is assembled as follows: Section II overviews AVC /H.264 Standard and CAVLC decoder. Section III provides a basic performance analysis of CAVLC Decode. Section IV concludes the paper.

## II. OVERVIEW OF AVC/H.264 STANDARD AND CAVLC DECODER

Existing VC standards that have been extensively employed in multimedia applications over time include MPEG-2 [1], VP8 [2], and H.263 [3]. These standards, however, have slow speeds of video compression and decompression. Scenes with quick motion, occlusions, or intricate motion patterns may be complex for traditional motion compensation. While motion correction can be designed to function well on devices with low processing power, these standards must have high computational complexity to achieve extraordinary compression efficiency.

### *A. AVC/H.264*

The goal of the new H.264 video coding standard is to increase significantly the effectiveness of vertical cabling compared to the AVC/H.264 standard. Over its predecessor, it achieves a 50% gain in coding efficiency. The block diagram of the AVC decoder is shown in Fig. 1. First, the entropy decoder block processes the compressed input data, which extracts the different syntax elements (SE). The residuals are then inversely quantized and transformed by the inverse quantization and inverse transform blocks. Block prediction is used intra- and interframe, depending on the bitstream's input value [12]. When making a prediction, intra-prediction uses pixels from the same block, but inter-prediction uses pixels from other blocks.

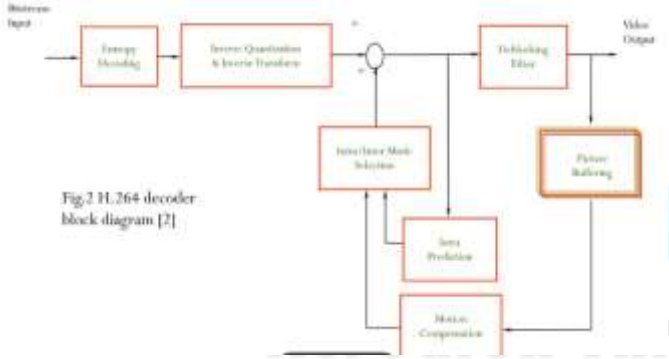


Fig. 1. AVC decoder block diagram.

Utilizing the pixels from the reference blocks, predictions are made. Conventional block-based hybrid coding partitioning and quantization processes can produce coding artifacts such as block discontinuities, ringing artifacts, mosquito noise, or texture and edge smoothing [13]. In-loop filtering techniques are employed throughout the encoding and decoding procedures to reduce these artifacts. To take advantage of spatial redundancy and increase compression efficiency, the Intra Prediction Mode (IPM), Sample Adaptive Offset (SAO), and Deblocking Filter (DBF) use different intra-prediction modes within each frame. (RPS) Selection of reference images: Selecting reference images carefully can assist in minimizing artifacts and enhancing overall visual quality.

### *B. CAVLC DECODER*

The entropy coder, often referred to as a CAVLC decoder, is the first part of an AVC decoder, as illustrated in Fig. 1. Entropy decoding, context modeling unit, inverse mapping module, reconstruction logic, error handling technique, and video frame reconstruction are the three main components of CAVLC. The CAVLC decoder uses bitstream processing to decode the bits that are encoded at variable lengths, reconstruct the original syntax parts, and then use those reconstructions to recreate the decoded video frames.

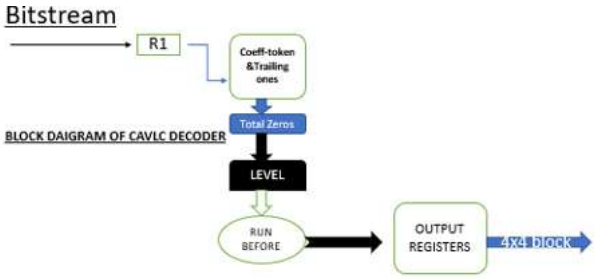


Fig. 2. CAVLC decoder.

It transforms an input stream of bits into several regular and bypass bins. Token coefficients, like motion vector components, transform coefficients in a block of video data or other syntax parts and represent real data. "Trailing ones" are the term used to describe the subsequent non-zero coefficients after a block of transform coefficients in video compression. Based on the decoded data and the size of the coefficient block, the total zeros function would compute this value: Total Zeros=TotalCoeff−1−NonZeroCount. The run-before function is essential to accomplishing effective video data compression by compactly expressing the position of non-zero coefficients. It increases the overall compression efficiency of algorithms and aids in the reduction of redundancy in the encoding process.

By managing the timing, integrity, and flow of data signals, output registers are essential to digital systems' correct functioning and dependability. This is separated into regions for rLPS and rMPS. The width of these zones is rLPS=range X pLPS. This formula determines the breadth of the LPS (Less Probable Symbol) zone. Range X (1-pLPS) = rMPS The breadth of the MPS (More Probable Symbol) is determined by this formula. The probability of a symbol being an MPS is typically higher than that of it being an LPS.

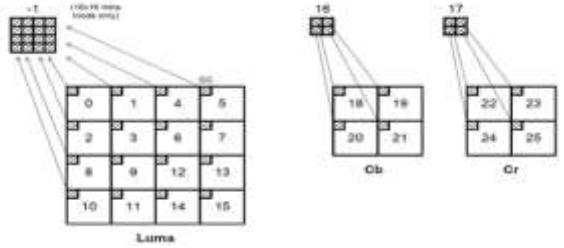


Fig.3 coding order block in macroblocks

Using the decoded values, the CAVLC decoder can be further broken down into how the decompression process functions. The syntax elements are reconstructed.

**Entropy decoding:** After extracting the variable-length encoded bits from the compressed bitstream, the CAVLC decoder starts by entropy decoding them. The original syntactic pieces are recovered by entropy decoding, which is the opposite of the variable-length encoding technique used during compression.

**Context modeling**: by capturing statistical relationships between symbols and their settings, context modeling, a crucial part of entropy coding in video compression, enables effective representation of video data. It is essential to maintaining video quality and reaching excellent compression efficiency.

**Inverse mapping**: the CAVLC decoder uses inverse mapping to reconstitute the compressed video data for playback or additional processing by reassembling the decoded variable-length codes into their original grammar pieces. Reverse mapping must be implemented effectively to provide high-speed and high-quality video decoding.

**Reconstruction logic**: the CAVLC decoder uses reconstruction logic to reassemble the decoded syntax pieces into video frames prepared for viewing or additional processing. Post-processing techniques, inverse transformation, and inverse quantization are used to recover the original video data from the compressed bitstream.

**Error handling:** To ensure dependable and resilient video decoding in various operating situations, a CAVLC decoder uses a combination of error detection, resilience approaches, feedback mechanisms, and concealment strategies.

**Video frame reconstruction**: before finishing the decoding process, the last steps entail creating the output frames, recreating the video frames from the decoded syntax elements, and optionally reporting any remaining faults.

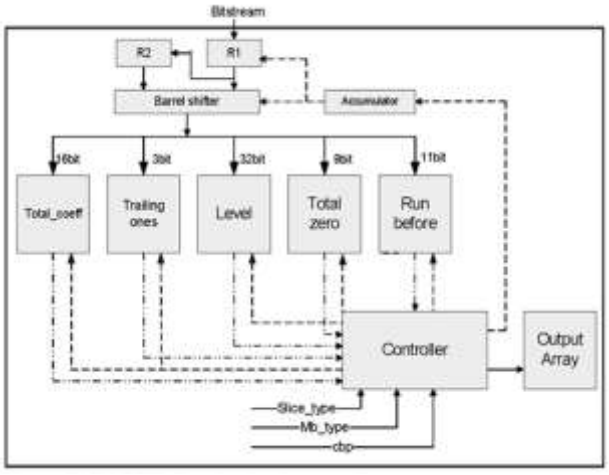


Fig.4 CAVLC Decoder hardware architecture

## III. ANALYSIS OF PERFORMANCE BY USING CAVLC DECODE

In the context of the AVC standard, entropy coding techniques frequently include CAVLC to compress video data effectively. This entropy coding technique is one of the two that the H.264 and H.265 standards support; the other one is called CABAC (Context Adaptive Binary Arithmetic Coding). It is both the continuation and the most advanced version of CAVLC coding. While HEVC is used in CABAC and can be used for 4k (4096x2160 pixels) video coding, AVC is used in CAVLC for HD (1920x1080 pixels). Although CABAC typically provides stronger robustness and compression efficiency, it does so at the expense of more computational complexity. AVC coding standard may lose some compression efficiency in favor of CAVLC, which is more straightforward and computationally efficient.

The project's main objective is to use the FFMPEG codec to speed up the CAVLC decoding process without sacrificing the quality of the source video. When decompressing, utilize the video bitstream FFMPEG decoded and incorporate it into the Verilog code to maintain the video's original quality. Its scalability, real-time processing, energy efficiency, and performance advantages are dedicated.

A major obstacle to attaining quick and effective video decoding is the low computing complexity of CAVLC decoding. The current hardware-based CAVLC decoder frequently needs help to process HD video streams in real time.

To overcome this difficulty and enable HD video streaming in real-time, the project will design and divide pixel values using a method that can be done accurately and without requiring minute adjustments to the CAVLC decoding process. To investigate the FFMPEG CAVLC decoder, we must first create the video bitstream from the captured video to begin the extraction process.

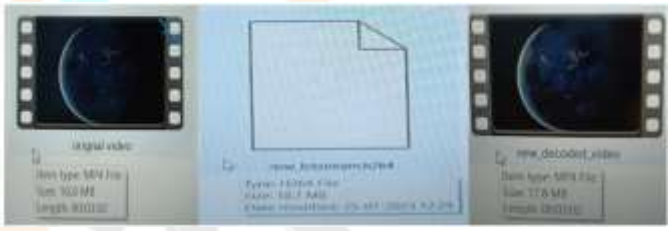

Fig.5 Generation and extraction of bitstream using FFMPEG

Codes for video encoding and decoding using FFMPEG

1}ffmpeg -i input.mp4 -c:v libx264 -crf 23 -preset medium -c:a aac -b:a 128k output.mp4

2} ffmpeg -i input.mp4 -c:v libx264 -preset medium -crf 23 -c:a aac -b:a 128k output.mp4

3} ffmpeg -i input.mp4 -c:v libx264 -preset medium -tune zerolatency -x264-params keyint=60:min-keyint=60 -an output.h264

4} ffmpeg -i input.mp4 output.av

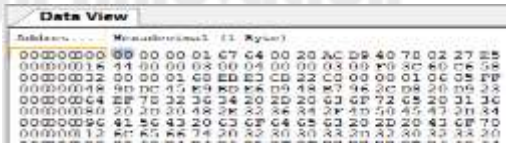


Fig.6 view of the syntax elements and bitstream extraction

Fig.7 Generated output using Verilog code by compiler

To study the CAVLC decoder, we use the Verilog coding of the generated bitstream so that the decompression of the video coding shall not cause the video quality to decrease. After decoding the video, this writes the CAVLC top code by taking the bitstream as input and writing a code to give the output as the syntax element.

## IV. Conclusion

This paper describes the CAVLC Decoder video decompression from the compressed bitstream using H.264/AVC video coding standards using FFMPEG and Verilog coding. We have completed the bitstream extracting and getting the syntax elements of the CAVLC Decoder for the AVC video coding standard, resulting in the generation of decoded video. Thus, the syntax elements produced by decoded video using Verilog work well and compile a good quality video per the required criteria. Therefore, in the future, using the same video, we will upscale the video frames in terms of performance, functionality, and integration with other parts, which may all be tested and verified in real-time using an FPGA board.

## References


[1] "Information technology – generic coding of moving pictures and associated audio information: Video," 2013. [Online]. Available: https://www.itu.int/rec/T-REC-H.262

[2] "Advanced video coding for generic audiovisual services, itu-t, itut recommendation h.264, august 2021.-h.264, year=2021, volume= , number= , pages= , url=https://www.itu.int/rec/T-REC-H.264-202108-I."

[3] "H series: Audiovisual and multimedia systems h.200-h.499: Infrastructure of audiovisual services h.260-h.279: Coding of moving video," 2001. [Online]. Available: https://handle.itu.int/11.1002/1000/5665

[4] "High-efficiency video coding, document rec. itu-t h.265 v8 and iso/iec 23008-2 (twinned)," 2021. [Online]. Available: https://handle.itu.int/11.1002/1000/14660

[5] W. Yu and Y. He, "A high performance cavlc +

[6] decoding architecture," *IEEE Transactions on Consumer Electronics*, vol. 51, no. 4, pp. 1352– 1359, 2005.

[7] G. J. Sullivan, 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*, vol. 22, no. 12, pp. 1649–1668, 2012.

[8] Y.-H. Chen and V. Sze, "A deeply pipelined cabac decoder for hevc supporting level 6.2 high-tier applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 5, pp. 856–868, 2015.

[9] 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, 2003.

[10] V. Sze and D. Marpe, "Entropy coding in hevc," *Springer International Publishing*, pp. 209–274, 2014.

[11] J. Zhou, D. Zhou, S. Zhang, S. Kimura, and S. Goto, "A variable-clockcycle-path vlsi design of binary arithmetic decoder for h.265/hevc," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 2, pp. 556–560, 2018.

[12] J.-W. Chen, 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*, vol. 56, no. 4, pp. 2529–2536, 2010.

[13] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h.264/avc video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[14] M. Karczewicz, N. Hu, J. Taquet, C.-Y. Chen, K. Misra, K. Andersson, P. Yin, T. Lu, E. Franc¸ois, and J. Chen, "Vvc in-loop filters," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3907–3925, 2021.

[15] 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, 2003.

[16] R. Osorio and J. Bruguera, "Arithmetic coding architecture for h.264/avc cabac compression system," in *Euromicro Symposium on Digital System Design, 2004. DSD 2004.*, 2004, pp. 62–69.

[17] A. Petrovsky, A. Stankevich, and A. Petrovskyi, "Pipeline processing in real-time of cabac decoder based on fpga," in *2012 International Conference on Signals and Electronic Systems (ICSES)*, 2012, pp. 1–4.

[18] K. Won and B. Jeon, "Complexity-efficient rate estimation for mode decision of the hevc encoder," *IEEE Transactions on Broadcasting*, vol. 61, no. 3, pp. 425–435, 2015.