International Conference on Control, Engineering & Information Technology (CEIT'13)
Proceedings Engineering & Technology - Vol.3, pp. 216-218, 2013

A high level Hardware Architecture Binarizer for H.264/AVC CABAC Encoder

N. Jarray, S. Dhahri, M. Elhaji and A. Zitouni
Electronic and Micro-Electronic Laboratory, Faculty of Sciences of Monastir, Monastir 5000, Tunisia
[jarray.nedra, dahrisalah, abdelkrim_zit]@yahoo.fr

***Abstract*—The standard H.264/AVC represents an enormous step forward in the field of the technologies of video compression. It guarantees a better efficiency of compression because of the biggest precision of the predictive functions and the better tolerance of the errors, also provides new possibilities for the creation of video encoders offering video flows of higher quality, frequency of image more important and higher resolution. In the standard H.264, there are two entropy coders which are distinguished from the other video standards, CAVLC (Context Adaptive Variable Length Coding) and CABAC (Context Adaptive Binary Arithmetic Coding). In this paper, we present the various blocks binarization, context-model and arithmetical binary coding) of the encoder CABAC. In addition, a new hard architecture of binarization process of CABAC has been proposed which supports all binarization techniques. These results were achieved by using the ModelSim tool for the simulation and the ISE tool for the synthesis targeting the FPGA technology of Xilinx. The proposed design consumes about 400 slices and works on the 145.15 MHz frequency.**

I. INTRODUCTION

H.264/AVC is the newest international video coding standard. It has been adapted as videophone, HDTV, digital Multimedia broadcasting. This norm provides much higher performance than the last standards. It consists of many advanced compression techniques such as Integer DCT, Quantification, and Prediction Intra/inter frame and Entropy coding [5].

H.264/AVC employs two kinds of entropy coding, Context-based Adaptative Variable length Coding (CAVLC) and Context Adaptative Binary Arithmetic Coding (CABAC). CABAC is a shape of entropy coding used in H.264/AVC video coding. It is a lossless compression technique, in addition it is lonely supported in Main and Higher profiles and needs a large amount of processing to decode compared to similar algorithms. Also, CAVLC [4] is a form of entropy coding used in H.264/AVC; it is a lossless compression technique. Unlike CABAC, CAVLC is supported in all H.264 profiles and is used to encode only the residual coefficient.

In comparison with CAVLC, CABAC is more effective, achieves an average more important between 9% -14% of bit rate saving CAVLC.

In literature, much work can be found showing the CABAC architecture. However, most of the work doesn't describe well enough the processing stages of CABAC. Nonetheless, there are some works on hardware design that present a binarization process. Similarly in [1] the architecture proposed supports all techniques of binarization defined in H.264/AVC. In [2] the architecture suggested implementing the function of binarization and context index for each bin in parallel.

The rest of this paper is organized as follows: in part 2, we present the CABAC algorithm, and then in part 3, we introduce the proposed architecture, later the implement results and comparisons are presented in part 4.

II. THE CABAC ALGORITHM

The CABAC encoder block Diagram is presented in Figure1. It consists of three steps: Binarization, Context modeling and Arithmetic coding.

Figure1: Encoder block Diagram.

- BINARISATION

The binarisation [3] process translates a non_binary syntax element (SE) (defined in H.264/AVC standard) to a bin string. In CABAC, there are four basic methods: unary code, truncated unary code, fixed length code and exp-Golomb code are defined to binarize most of the SE. In addition, some of SE concatenation of the basic binarization schemas are used and for the SE of MBTYP and SMBPTY is binarized by look up table.

- CONTEXT MODELING

The context modeler [3] reads in the bin string and generates context value according to neighboring data of the top and left macro bloc. The value context is an index to the context table built at the beginning of processing a new slice.

One of the most important properties of arithmetic coding is the possibility to utilise a clean interface between modeling and coding, so that in the modeling steps, a model probability distribution is assigned to the given symbols, which then in the coding stage drives the

actual engine to generate a sequence of bit as a coded representation of the symbol according to the modeling distribution. In CABAC, there are four basic design types of context models which can be distinguished. The first type involves a context template with up to neighboring syntax elements in the past of the curent syntaxe element to encode.The second type of context models is defined for the syntax element of mb-type and sub-mb-type. Finally both the third and fourth type of context model is applied to residual coefficient.

- ARITHMETIC ENCODING

The binary arithmetic [4] is based on the principle of recursive interval subdivision that involves the following elementary multiplication operation. In binary arithmetic coding tow types of symbols; Least Probable Symbols (LPS) and Most Probable Symbols (MPS) are coded rather of '0' and '1'. LPS has an estimated probability namely pLPS and pMPS respectively.

The arithmetic encoder module consists of three coding engines; the regular coding engine, the by-pass coding engine and the termination coding engine.

III. THE PROPOSED ARCHITECTURE

The binaization method consists of mapping the integer value of (SE) into a sequence of bit that represents the original value called bin string. The size of the bin string generated for each input depends on the types of SE.

This process is done to reduce the size of the symbol coded; hence it simplifies the costs of context modeling as well as the calculus of arithmetic coding.

To implement the binary process, CABAC defines four methods such as Unary (U), Truncated Unary (TU), Fixed Length (FL), Unary/Kth Exp-Golomb (UEGk). The selection among these different techniques will be used depending on the SE types, the macro bloc type, slice type and the value of SE. The architecture of binarisation proposed is in figure 2. It is a little different from the other related work. It is composed of a demultiplexer, a multiplex and the same technique of binarization was used. The object of this section is to provide an evaluation of performance of the binarisation bloc in terms of area and clock frequency.

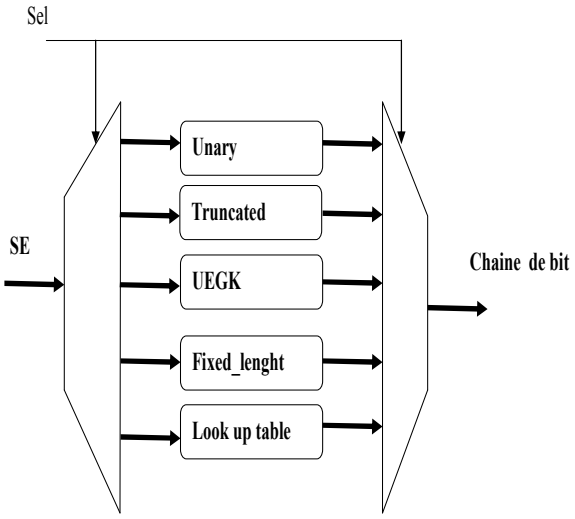

Figure 2: Binarization architecture.

Unary: translate the unsigned integer value in the string of one ('1') with size equal to the value of SE coded and terminated by zero ('0').

Truncated Unary: the Unary code used when the valSE is lower than Cmax and if valSE equal to cMAX, the terminating '0' is ignored.

Fixed length: the codeword of valSE equal to the valSE bit with length FLbits=log2 (cMax+1) bits.

Table 1: Example of U, TU, FL binarization process.

SE	BIN		
	Unary	Truncated Unary	Fix length
0	0	0	0000
1	10	10	0001
2	110	110	0010
3	1110	1110	0011
4	11110	1111	0100

Unary/kth Exp-Golomb: is the concatenation of prefix and suffix bit string. The prefix part it is calculated using the truncated unary method. For the suffix part is determined by {1, suffix} =| coeff_level – 14|

Table 2: Example of KthEG technique binarization.

|Level|	Truncated Unary, S=14	Kth EG with k=O		
19	11 1111 1111 1111	prefix	\	suffix
		11	0	01

- BINARISATION PROCESS OF MB_TYPE AND SUB_MB_TYPE

The SE of mb_type and sub_mb_type are binarized by looking up table. The selection of which kind of table used depends on the type of prediction mode, slice and frame of macro block given. If the prediction mode is inter frame, the bin string is generated by prefix and suffix from two different tables that can be concatenated to produce the bin string. Much detail can be found in the H.264/AVC standard.

IV. RESULTS AND COMPARISON

The proposed binarization architecture has been described in VHDL and implemented in a Vertex IIP of the Xilinx FPGA. The RTL synthesis results of the binarization architecture are shown in the table 3.

Table 3: Synthesis results

N. Slice	N. DFF	N. LUT	F. MHz
400	431	751	145.15

In table 4 we present a comparison of our architecture with other works. We cannot find many works of binarization process which work with FPGA, thus we choose these works [1] and [6] because the FPGA were used. Table 4 shows that our design saves 21.5% hard resource and is better in terms of frequency relative than [1]. Our work also gives a better result in terms of hard resource relative [6].

Table 4: Comparison with related works.

works	Function	Technologies	Area(slice)	F.MHz)
[6]	binarization	FPGA	403	---
[1]	binarization	FPGA	509	142.4
Our work	binarization	FPGA	400	145.15

V. CONCLUSION

In this paper, we presented the various blocks (binarisation, context-model and arithmetical binary coding) of the encoder CABAC. In addition, a new hard architecture of binarisation process of CABAC has been proposed which supports all binarisation techniques. Our aim is to supply an assessment of performance of the binarisation block in terms of area and clock frequency. The modeling of this block is followed by one or several results of simulation and synthesis so validating the functioning of this block.

We also made a comparative study which shows that our work is saving hardware resources compared with related works.

REFERENCES

[1] Andre Luis del Mestre Martins, Vagner Rosa, Sergio Bampi" A Low-Cost Hardware Architecture Binarizer Design for the H.264/ A VC CABAC Entropy Coding "Informatics Institute - PPGC - Federal University of Rio Grande do SuI Porto Alegre, Brazil ©20 10 IEEE.

[2] Yizhong Liu Tian Song Takashi Shimamoto "High performance binarizer for H.264/AVC CABACElectrical and Electronic "Engineering Graduate School of Engineering Tokushima University.

[3] Detlev Marpe, Heiko Schwarz, and Thomas Wiegand "Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard" IEEE. 7, JULY 2003.

[4] ITU_T, International Telecommunication Union "Advanced video coding for generic audiovisual services (lTU Rec. H.264 (E)/ISO/IEC 14496-10:2005 (E))," 2005.

[5] T. Wiegand, G. J. Sullivan, G. Bjontegaard, A. Luthra, "Overview of the H.264/AVC video coding Standard", IEEE Trans. Circuit and Systems for Video Technology, July 2003.

[6] Roberto R. Osorio and Javier D. Bruguera "High-Throughput Architecture for H.264/AVC CABAC Compression System ", IEEE 11, NOVEMBER 2006