Multi-shape Hardware Design for the Adaptive Loop Filter

Ruhan Conceição, Ândrio Araújo, Bruno Zatt, Marcelo Porto, Luciano Agostini
Group of Architecture and Integrated Circuits - GACI
Federal University of Pelotas – UFPEL
{radconceicao,afdacampos,zatt,porto,agostini}@inf.ufpel.edu.br

ABSTRACT

This paper presents a multi-shape hardware design for the Adaptive Loop Filter (ALF) which is responsible to reduce the distortion between an original image and the encoded image during the video coding process by fixing artifacts from previous stages. It was proposed a hardware design for the ALF core which is capable to process all ALF shapes proposed along the High Efficiency Video Coding (HEVC) standardization. This design aims to saving hardware resources through the use of a reuse approach and also aims to process UHD 4K (3840 x 2160 pixels) videos in real time at 60 frames per second. The synthesis process was targeted to Altera Cyclone II FPGA and ASIC 65nm technologies. The synthesis results show that the designed architecture is capable to process 60 UHD 4K frames per second, considering the ASIC implementation, with a power dissipation of only 10.04mW. Compared to related works, the proposed design presents gain in terms of both hardware resources usage and number of ALF shapes able to be processed.

Categories and Subject Descriptors

B.7.1 [Integrated Circuits]: Types and Design Styles – advanced technologies, algorithms implemented in hardware, VLSI (very large scale integration).

General Terms

Performance, Design.

Keywords

ALF; Hardware Design; HEVC; Multi-shape

1. INTRODUCTION

Along the last few years, not only quality and resolution of digital videos have been increasing in a fast and steady manner, but also the number of electronics devices that support these videos (smartphones, tablets, Blue-ray players, etc.), demanding the need to represent a huge volume of data. Generally, these devices are battery powered and contain several energy and hardware limitations. As consequence, the study and improvement of digital encoders/decoders have become an essential and important activity in the current scenario. Furthermore, those devices that process digital videos must be able to process high-resolution videos in real time – 30 or 60 frames per second, depending on the video resolution –, while maintaining low energy consumption and a low use of hardware resources. For this reason, researchers are constantly looking for improving video encoders/decoders in terms of compression rate, video quality, computation complexity and energy consumption.

Video coding is an essential element in applications that handle digital videos, since an uncompressed video requires a prohibitive volume of bits to its representation. On January 2010, experts from ITU-T and ISO/IEC founded the Joint Collaborative Team on Video Coding (JCT-VC) to develop a new video coding standard called High Efficiency Video Coding (HEVC) [1], with the goal of increasing the video compression in 50% while maintaining the same computer complexity, when compared to the former standard, the H.264/AVC [2]. When the HEVC standard was released it was not able to maintain the same computer complexity, increasing it compared to its predecessor (H.264/AVC) [3]. However, despite the complexity increasing, the HEVC achieved its goal in compression rate gain in comparison with H.264/AVC, presenting an average increase of 50% [4]. Moreover, this standard is efficiently capable to encode/decode digital videos with resolutions up to Ultra High Definition 8K (UHD 8K – 7680x4320 pixels), while the H.264/AVC focuses on video resolutions up to the HD 1080p (1920x1080 pixels) [4].

The HEVC consists of many stages to encode/decode digital videos such as intra-frame and inter-frame predictions, transforms, quantization and entropy coding stage [4]. However, during this process, the objective and subjective quality can be deteriorated, especially through the quantization stage, which insets artifacts in the video as a collateral effect to increase the compression rate. In this context, there are filters aiming to increase the quality of the encoded video. Initially, a set of three filters, called In-Loop Filter, were proposed to the HEVC standard. Deblocking Filter (DF) – in charge to reduce the block effect –, Sample Adaptive Offset Filter (SAO) – that aims to reduce the ringing artifacts – and the Adaptive Loop Filter (ALF) [4], the focus of this work, composed the first In-Loop Filter proposed to the HEVC. The ALF is responsible to reduce the coding error of output and reference (to be used as reference during the inter-frame prediction).

Research based on the HEVC Model (HM) software, conclude that ALF can achieve 4-5% bit rate reduction for High Definition video sequences when B-predictive frames are allowed and, 10% when only P-predictive frames are considered. However, the inclusion of the ALF rises the decoding time in 7-14% [5]. Thus, in spite of the coding gains, the complexity brought by this filter was considered too high. Due this fact, the ALF was removed from the HEVC project in the HM 8.0 version [6]. This decision, however, was not unanimous in the video coding community. Claiming the importance of the ALF coding gains, some researchers such as [7] and [8], keep defending the reincorporation of the ALF filter in the HEVC. For this reason, studies focusing the ALF are still relevant, since that, it can be incorporated in one of the HEVC extensions, as the 3D-HEVC [9], or in a new video coding standard.

The high number of mathematical operations to filter a sample, plus the current high-definition digital videos resolutions, requires a high computational effort to filter a video picture in real time. Due this fact, ALF became an ideal target for hardware implementations, as it is possible to see in related works as [10] and [11]. Rediess [10] proposes a solution with three different hardware designs, one for each size of the ALF diamond shape. In [11], our previous work, it is proposed a hardware design, which is capable to perform a multi-size solution for the work presented in [10]. More hardware design works targeting the ALF filtering process were not found in the literature.

Although [11] proposes a hardware solution for three different ALF shapes, along the HEVC development many shapes were proposed. For this reason, this work presents a hardware design capable to calculate a filtered sample for all ALF filter shapes (as it will be explained later), which were proposed along the HEVC standardization. The proposed architecture aims to achieve processing rate enough to process UHD 4K (3840x2160 pixels) videos in real time at 60 frames per second.

2. ADAPTIVE LOOP FILTER

The ALF was proposed during the development of the HEVC aiming to reduce the mean square error between original samples and decoded samples by using Wiener-based adaptive filter [5]. The filter is applied to the reconstructed image after the DF and SAO filters. Basically, it works to reduce the distortion error generated by the previous coding modules.

In general terms, two stages compose the whole ALF filtering process. The first one evaluates the filter coefficients whereas the second one is responsible for the application of the filter into the decoded samples. The focus of this work is in the sample-filtering process. More details about the whole ALF process can be found at [5].

In order to filter a given sample, the ALF also uses in its calculations other neighbor samples together, constituting a filter shape. Along the HEVC development, many filter shapes were proposed looking for reducing the number of necessary calculations to perform the ALF while keeping an acceptable filter performance.

The proposed filter shapes along the HEVC definition are presented in Fig. 1. In the first ALF versions, the filter shape was in diamond format with sizes 5x5, 7x7 and 9x9 (Fig. 1 (a), (b), (c), respectively) [5]. The diamond shape reduced half of coefficients needed, in comparison with a square set of samples with the same height and length of the filter in discussion. Fig. 1 also shows other two different shapes: square (Fig. 1 (d) and (e)), and the 9x7 cross shape (Fig. 1 (f)). Although the filter shape presented in Fig. 1 (c) has seven samples in the vertical, it is called 9x9 even so.

In order to demonstrate how a sample is filtered, Fig. 1 (d) and (e) shows the ALF-Square 5x5 shape. Fig. 1 (d) illustrates the pixels samples that will be used in the filtering process to generate the new value for the sample *a'*. Moreover, Fig. 1 (e) shows the coefficients that will be working in the filtering process. The filter process corresponds to a multiplication of the sample with its corresponding coefficient, after that, the results are added generating only one sample. Equation (1) illustrate how the filtered sample *a'* is generated considering the 5x5 ALF square-shape shown in Fig. 1 (d) and (e).

$$a' = a*C0 + b*C1 + c*C1 + \cdots + p*C8 + q*C8 \qquad (1)$$

Usually the ALF is not applied in all video picture samples due the coding parameters and video characteristics. This way, when the ALF is applied over all samples, it becomes a worst-case scenario, where it is required the higher computational effort.

3. PROPOSED OPTIMIZATION AND HARDWARE DESIGN

This section presents the developed optimization, which aims to integrate all ALF filter shapes. This optimization was done targeting the development of a low-cost design, in terms of hardware resources consumption.

3.1 Proposed Optimization

To develop the optimization, it was analyzed the relation between the different filter shapes and which operations can be shared among the different shapes. Since ALF-DS 9x9 (diamond shape) is the biggest shape – in other words, this shape uses the higher number of samples and coefficients to filter a sample among all others shapes discussed in this paper –, it allowed the derivation of all the other filters shapes from the ALF-DS 9x9, which was developed after a thorough analysis.

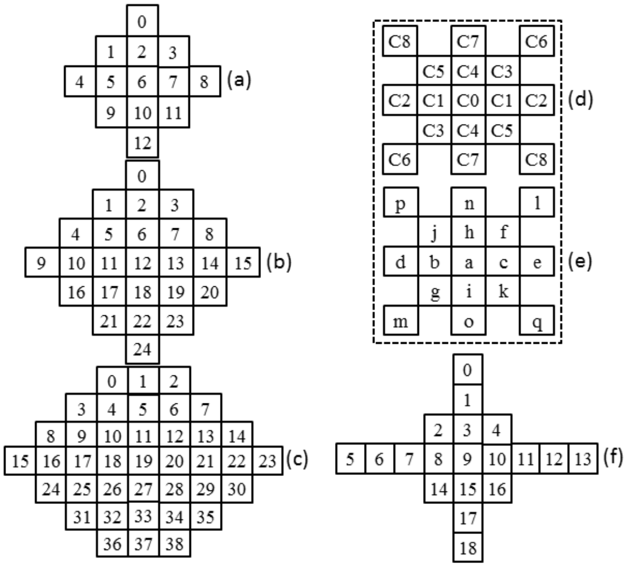

Figure 1. ALF filtering shapes, (a) square shape (coefficients), (b) square shape (samples), (c) 5x5 diamond shape, (d) 7x7 diamond shape, (e) 9x9 diamond shape, (f) 9x7 cross shape

Figure 2 present a derivation structure of the different filters shapes, where, sample sets can be discarded in order to use the desired filter shape. The sample sets are the *Core* (C), *D5* (diamond), *D7A*, *D7B*, *D9A*, *D9B* and *Square* (S). For an example, the set *D7* means diamond 7x7, because, together with the sets, *Core* (common to all the shapes) and the *D5* (diamond 5x5), it will form the ALF-DS 7x7. It is important to observe, that the *D7* set has two different categories, *A* and *B* (*D7A and D7B*). This occurs because the ALF-Cross 9x7 filter shape needs to process only the *D7A* set and does not need use the *D7B.* To choose the desired filter shape, the set of samples to be processed needs to be turned ON or OFF as shown in Table I.

3.2 ALF-MS Hardware Design

Three main modules compose the full ALF-MS (multi-shape) architecture: *control*, *filtering module* and *clip*, as shown in Fig. 3. The input of the architecture is 39 8-bit samples (amounting to 312 bits), 20 10-bit filter coefficients (amounting to 200 bits) and three bits to select the ALF filter shape.

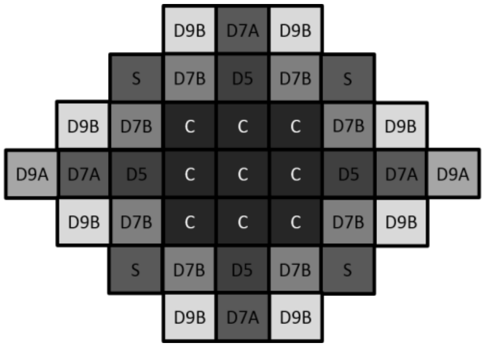

Figure 2. ALF derivation structure

Table 1. Samples set configurations

SHAPE ALF	Samples Set						
	C	**D5**	**S**	**D7A**	**D7B**	**D9A**	**D9B**
DS 5x5	ON	ON	OFF	OFF	OFF	OFF	OFF
DS 7x7	ON	ON	OFF	ON	ON	OFF	OFF
DS 9x9	ON	ON	ON	ON	ON	ON	ON
Square 5x5	ON	ON	ON	OFF	OFF	OFF	OFF
Cross 9x7	ON	ON	OFF	ON	OFF	ON	OFF

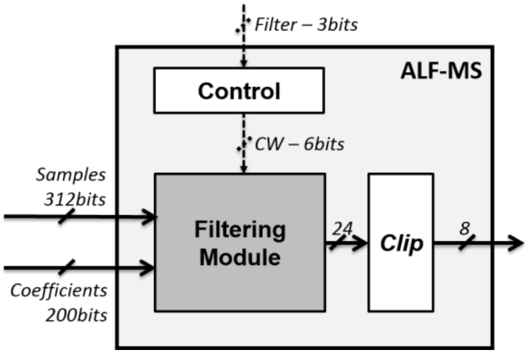

Figure 3. ALF-MS block diagram

3.2.1 *Filtering Module*

Since the process to calculate the filtered sample is performed through a sum of multiplications, a set of sums and multiplications can be performed separated and be added later. The filtering module uses this property to perform the selected ALF-shape, discarding the sums and multiplications that the filter does not use for the selected shape.

The filtering module is composed by seven filtering units, which process the desirable filter shape (accordingly to the samples set configuration presented in Table I). Fig. 4 presents a block diagram of this module. It is important to highlight that the filtering units order is different in Fig. 4 compared to the Table I. It was done in order to make Fig. 4 more readable.

Each filtering unit is an ALF-N hardware design, where N means the number of input samples used to perform the calculations. This variable N can be “2”, “4”, “8” or “9”, according to the unit size. Fig. 5 presents the block diagram of the ALF-4 hardware design. The other filtering units follow the same idea, eventually using less or more number of samples and coefficients as input.

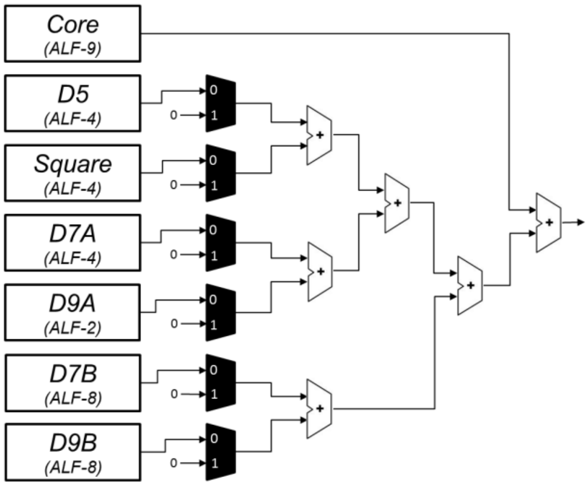

Figure 4. Filtering module block diagram

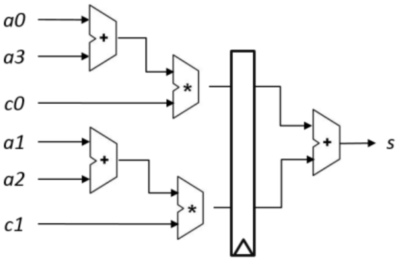

Figure 5. ALF-4 hardware design block diagram

3.2.2 *Control Module*

The control module converts the selected filter shape at the hardware input to the control word (CW), which defines the value of the multiplexers used in the Filtering Module. Since all filter shapes use the Core unit, there is no need to implement a multiplexer for this unit; hence, the CW has 6 bits instead of 7 bits. Finally, for design issues, the filtering unit is enable when its corresponding bit is set to “0” and disabled when “1”.

3.2.3 *Clip Stage*

The filtered sample is represented in 24 bits after the filtering module, which is different to the HEVC bit depth (8 bits). Thus, a clip operation is performed to normalize the output in order to readapt the result to the pixel range, between 0 and 255, where the 16 less significant bits are removed.

4. SYNTHESIS RESULTS

The proposed architecture was described in VHDL, using the Altera Quartus II software tool and synthesized targeting an ASIC implementation, using a 65nm TSMC standard-cell technology with the Synopsys DC-Compiler tool. In order to compare with related works, this implementation was also synthesized targeting an Altera Cyclone II EP2C70F896C6 FPGA device.

Table II shows the synthesis results of the hardware design targeting the ASIC implementation. Number of frames processed per second is given considering only luminance samples. The performance calculation is always performed targeting the worst-case scenario, when all the video samples must be filtered. The

ASIC synthesis was generated considering three different operational frequencies, targeting the performance of HD 1080p@30fps and UHD 4K@30/60fps.

Table 2. ALF-MS ASIC Synthesis Results

Video Resolution	Parameter		
	Gates	Frequency	Power
HD 1080p@30fps	17,846	62.21 MHz	1.55 mW
UHD 4K@30fps	18,240	248.83 MHz	5.01 mW
UHD 4K@60fps	18,869	497.67 MHz	10.04 mW

It is possible to notice that the ALF-MS hardware design is capable to process 60 UHD 4K (3840 x 1920 pixels) frames per second, with a power dissipation of only 10.04mW and 5.01mW when processing 30 frames per second. Moreover, when considering HD 1080p applications, the power dissipation decreases to 1.55mW, considering the processing of 30 frames per second.

Since related works were synthesized in an Altera Cyclone II device, the hardware design presented in this work was also synthesized in this technology. Table III shows the comparison of this work with related ones.

Table 3. FPGA Results and Comparison

Parameter	Implementation		
	ALF-MS	[10]	[11]
Logic Elements (ALUTs)	919	2,071	1,660
Registers	532	1,553	1,028
DSP Blocks	40	22	11
Filter Shapes	5	3	3

As it is possible to observe, the developed design processes more ALF filter shapes (five) in a unique hardware design when compared to the related works (three). Moreover, even processing more filter shapes, it uses less hardware resources in terms of logic elements – given in terms Adaptive Look-up Tables (ALUTs) – and in terms of registers. This occurs because this design implements only one pipeline barrier, after the multipliers, whereas related works uses more (from seven to nine, depending the design) after the adders as well. Since adders are significantly faster than multipliers, there is no need to implement pipeline barriers after them.

This design has only a higher number of Digital Signal Processing (DSP) blocks, which implement the embedded multipliers. This occurs because we implement more filter shapes than related works.

Finally, it is important to highlight that [10] proposes three separated hardware designs for the ALF-DS filter shapes. This way, the hardware resources consumption of those three architectures were added in order to allow the comparison.

5. CONCLUSION

This work presented a multi-shape hardware design for the Adaptive Loop Filter (ALF-MS). The developed design is capable to process all the five different filter shapes proposed along the HEVC standardization in a single multi-shape hardware design. Synthesis results showed that our solution is capable to achieve real time processing on UHD 4K videos, processing 60 frames per second with a power dissipation of only 10.04mW. Moreover, when targeting HD 1080p@30fps videos, the developed hardware design dissipates only 1.55mW.

Comparing to related works, our solution is the only one to process all ALF shapes proposed in the HEVC standardization. In addition, the ALF-MS also uses less logic elements and registers than the related works compared. This occurs due the fact that our solution does not implement unnecessary pipeline barriers after the adders. As future work, we plan to implement all ALF structure, combining the sample filtering with the coefficients generation through the Wiener Filter.

6. ACKNOWLEDGEMENTS

Our thanks to CAPES, CNPq and FAPERGS for the financial support, allowing the development of this work.

7. REFERENCES

[1] JCT-VC Editors, Recommendation ITU-T H.265 – *High Efficiency Video Coding* (ITU-T Rec.H.265), April 2013.

[2] JVT Editors (T. Wiegand, G. Sullivan, A. Luthra), Draft ITU-T Recommendation and final draft international standard of joint video specification (ITU-T Rec.H.264|ISO/IEC 14496-10 AVC), 2003.

[3] F. Bossen, et al, "*HEVC Complexity and Implementation Analysis*", IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1685–1696, Dec. 2012.

[4] G. J. Sullivan, et al, "*Overview of the High Efficiency Video Coding (HEVC) standard*", IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1648–1667, Dec. 2012.

[5] C. Tsai, et-al., "*Adaptive Loop Filtering for Video Coding*", IEEE Journal of Selected Topics in Signal Processing, vol. PP, July 2013.

[6] Il-Koo Kim, et-al. HM8: High Efficiency Video Coding (HEVC) Test Model 8 Encoder Description. JCTVC-J1002. 10th JCT-VC Meeting. Stockholm, 2012.

[7] I. Chong, M. Karczewicz. AHG6: ALF in HM80. JCTVC-K0273. 11th JCT-VC Meeting. Shangai, 2012.

[8] C. Chen, et-al, AHG6: Further cleanups and simplifications of the ALF in JCTVC-J0048. JCTVC-J0390. 10th JCT-VC Meeting. Stockholm, 2012.

[9] Muller, K.; et al "3D High-Efficiency Video Coding for Multi-View Video and Depth Data" IEEE Transactions on Image Processing, 2013.

[10] F. Rediess, et-al., "High Throughput Hardware Design for the Adaptive Loop Filter of the Emerging HEVC Video Coding" in *25th Symposium on Integrated Circuits and Systems Design* (SBCCI), Brasília, Brazil, 2012.

[11] R. Conceição, et-al, "Configurable Hardware Design for the HEVC-Based Adaptive Loop Filter" in 5th Latin American Symposium on Circuits and Systems (LASCAS), Santiago, Chile, 2014.

[12] T. Wiegand, et-al. WD5: Working Draft 5 of High-Efficiency Video Coding. JCTVC-G001. 7th JCT-VC Meeting. Geneva, 2011.