FPGA IMPLEMENTATION OF LOW COMPLEXITY DE-BLOCKING FILTER FOR H.264 COMPRESSION STANDARD

S.Nisha[1] (nishasubu94@gmail.com), PG Scholar,Gnanamani College of Technology.
Mr.E.Sathishkumar M.E.,(Ph.D),Assistant Professor,Gnanamani College of Technology.

ABSTRACT
In this paper, weproposed a novel modified filtering order and boundary strength values to reduce memory and computational complexity of h.264 deblocking filter.The standard h.264 deblocking filtering order is that the vertical borders of the luminance and chrominance blocks are all filtered before the horizontal borders. Since the results of the vertical filtering are used in the horizontal filtering, these intermediate results must be stored. This processing order is expensive in terms of memory use. This proposed filtering order reduces the amount of memory needed for filtering process. This work is designed in verilog HDL and synthesized for Virtex 6 FPGA to compute the power consumption, LUT, slice registers. In proposed method, the FPGA performance parameters such as power consumption, slice LUT, slice registers will be minimized.

I Introduction

The new video coding standard Recommendation H.264 of ITU-T also known as ISO/IEC 14496-10 or MPEG4 Part 10 Advanced Video Coding (AVC), significantly outperforms the previous one (H.263) in bit-rate reduction. Deblocking filter is one of the functional blocks in H.264/AVC.The operation of the deblocking filter is the most time consuming parts of H.264/AVC video decoder. The block-based structure of the H.264/AVC architecture produces artifacts known as blocking artifacts. These blocking artifacts can occur from both quantization of the transform coefficients and block based motion compensation.In order to reduce blocking artifacts and improve compression efficiency, H.264/AVC standard employs the deblocking filter.Each video frames are divided into 16x16 pixels blocks called macroblocks. The deblocking filter is applied to all the edges of 4x4 pixels blocks in each macroblock except the edges on the boundary of a frame. For each block, vertical edges are filtered from left to right first, and then horizontal edges are filtered from top to bottom. This ordering introduces a dependency which has to be addressed for fast processing in high throughput applications. The decoding process is repeated for all the macroblocks in a frame.Several architectures have been proposed for the deblocking filter.

II. Literature Review

A. Norkin *et al.* [11] have described the in-loop de-blocking filter used in the upcoming High Efficiency Video Coding (HEVC) standard to reduce visible artifacts at block boundaries. The de-blocking filter performed detection of the artifacts at the coded block boundaries and attenuated them by applying a selected filter. Compared to the H.264/AVC de-blocking filter, the HEVC de-blocking filter has lower computational complexity and better parallel processing capabilities while still achieving significant reduction of the visual artifacts. M. Esche *et al.* [12] have introduced a new filter that combines both spatial and temporal information to provide subjective and objective quality improvement. The filter only requires a small overhead on slice level while using the temporal information conveyed in the bit stream to reconstruct the individual motion trajectory of every pixel in a frame at both encoder and decoder. This information was then used to perform pixel-wise adaptive motion-compensated temporal filtering. The filter performed better than the state-of-the-art codec H.264/AVC over a large range of sequences and bit rates. Also, the filter was compared with another, Wiener-based in-loop filtering approach and a complexity analysis of both algorithms was carried out. C. Karthikeyan and Dr. Rangacha [13] have used a new international standards H.264 for the compression of video images. This artifact reduced the picture quality of the reconstructed images and video. To improve the quality of the received picture De-blocking filters were used to remove the artifacts. They have introduced a De-blocking algorithm to remove the artifacts and also proposed the hardware implementation for same algorithm. To reduce the power consumption of hardware implementation a technique clock gating was introduced. Result showed that they achieved the result of 30% power reduction for clock gating technique at the cost of 2.3 % hardware and 5.8% clock speed.

III.Exisiting System

compression is the critical technology in today's multimedia systems. The limited transmission bandwidth or storage capacity for applications such as HDTV, video conferencing, 3G for mobile device, and Internet video streaming emphasizes the demand for higher video compression rates. To achieve this demand, the new video coding standard Recommendation H.264 of ITU-T [1], also known as International Standard 14496-10 or MPEG-4 Part 10 Advanced Video Coding (AVC) of ISO/IEC, has been developed. It significantly outperforms the previous ones (i.e., H.261 [2], MPEG-1 Video [3], MPEG-2 Video [4], H.263 [5], and MPEG-4 Visual or part 2 [6]) in bit-rate reduction. The functional blocks of H.264/AVC, as well as their features, are shown in Fig. 1. Comparing the H.264/AVC video coding tools (e.g., adaptive deblocking filter [7], integer DCT-like transform [8] instead of the DCT [9], multiple reference frame [10], new frame types (SP-frames and SI-frames) [11], further predictions using B-slices [12], quarter per motion compensation [13], or CABAC [14]) to the tools of previous video coding standard, H.264/AVC provides the most improved algorithm in the evolution of video coding as well as error robustness and network friendliness [15]–[20]. At the same time, preliminary studies [21] using software based on this new standard suggest that

H.264 offers up to 50% better compression than MPEG-2 and up to 30% better than H.263+ and MPEG-4 advanced simple profile. As our experimental result indicates, the

operation of the deblocking filter is the most time consuming part of H.264/AVC video decoder. The block-based structure of the H.264/AVC architecture produces artifacts known as blocking artifacts. These blocking artifacts can occur from both quantization of the transform coefficients and block-based motion compensation. In order to reduce the blocking artifacts, the overlapped block motion compensation (OBMC) [22] is adopted into H.263 standard. Unlike the OBMC in H.263, H.264/AVC adopts an adaptive deblocking filter [7] that has shown to be a more powerful tool in reducing artifacts and improving the video quality. As a result, the filter reduces the bit rate typically by 6%–9% while producing the same objective quality as the nonfiltered video [23]. As shown in Fig. 2(a) a nonfiltered and Fig. 2(b) a filtered image, we can observe the different image quality between the nonfiltered and the filtered image at QP equal to 36. Adaptive deblocking filter can also be used in interpicture prediction to improve the ability to predict other picture as well. Since it is within the motion compensation prediction loop, the deblocking filter is often referred to as an in-loop filter. DSP or SIMD computational architecture. In order to reduce the conditional branch operations, we include the content activity check operations, table-derived operations, filtering operations, and computation of boundary strength operations into the edge filtering operation (EFO) unit to accelerate the deblocking filtering of H.264/AVC video coding. In addition, we propose an efficient VLSI architecture to improve memory performance by four times when compared to previous designs. The proposed architecture is called configurable window-based architecture (WIN). It uses a novel processing order within a frame to simultaneously process the horizontal filtering of vertical edge and vertical filtering of horizontal edge. Hence, our architecture is able to significantly improve the system performance and reduce the power consumption in the embedded system. The organization of this paper is as follows. In Section II, the algorithm of the deblocking filter is given. Section III analyzes the computational complexity of H.264/AVC baseline decoder.

IV.Proposed System

In the H.264/AVC, the deblocking filter is applied after the inverse transform in encoder and decoder (before displaying macroblocks). The filter smoothes block boundary, improving appearance of decoded pictures. Here we modified the standard filter order and boundary strength values. The deblockingfillter algorithm used in the H.264 standard is more complex than the deblockingfillter algorithms used by previous standards. The standard h.264 deblocking filtering order isModified and implemented in this work. Modified filtering order is shown in fig 1. The filtering order looks like two vertical edges have been filtered, the horizontal edge can be filtered. While this horizontal edge is being filtered, a third vertical edge can be filtered. All luma pixels have to be filtered before filtering a Chroma pixel. If we take this in pipeline concept to allow Chroma pixels to enter the vertical filter while luma pixels are still present in the pipeline. This reduce the number slices and registers required for filtering. Furthermore, by using this filtering order the amount of memory needed for storing filtered 4 * 4 blocks is also reduced.

4.1. De-blocking filter process

The de-blocking filter reduces the blocking artifacts (visible discontinuities in the video) caused by block-based encoding with strong quantization. Filtering is applied separately for 4x4 blocks (so-called P and Q blocks), as shown in Fig.4.1

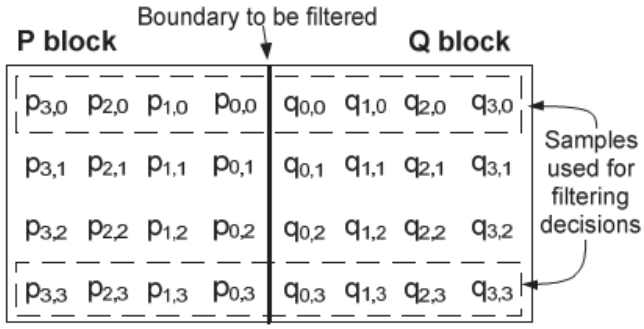

Fig.1. Boundary of a 4x4 block (blocks P and Q).

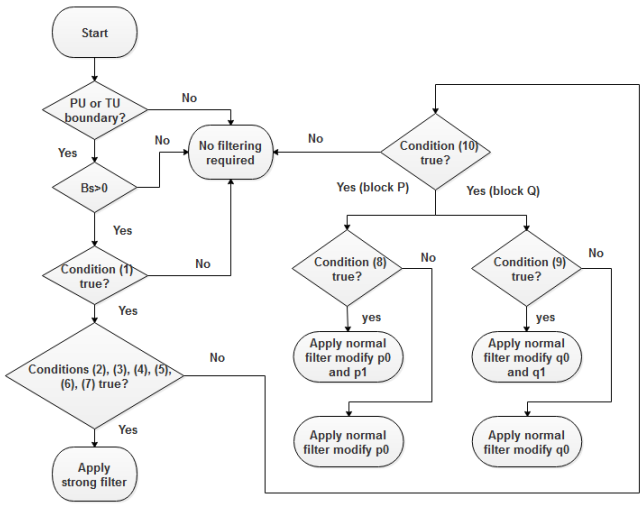

Fig: 2 Deblocking filter process flow

This unit determines the need of filtering two given 4x4 blocks. For input samples convention, please refer to Fig. 2. By examining the equations in Fig.5, it can be noted that conditions (1), (2), (3), (8) and (9) share similar equations and same input samples. Partial results from conditions (2) and (3) and used for conditions (1), (8) and (9). Hence, we employ hardware reuse to design a data path architecture for those conditions, as depicted in Fig.6. The condition equations require some multiplication by constants. We have replaced the multiplications by adders and shift operations to use less hardware resources. We have shorten conditions (1), (2), (3), (8) and (9) as c1, c2, c3, c8and c9, respectively.

Filtering decisions (conditions):

$$|p_{2,0} - 2p_{1,0} + p_{0,0}| + |p_{2,3} - 2p_{1,3} + p_{0,3}| + |q_{2,0} - 2q_{1,0} + q_{0,0}| + |q_{2,3} - 2q_{1,3} + q_{0,3}| > \beta \quad (1)$$

$$|p_{2,i} - 2p_{1,i} + p_{0,i}| + |q_{2,i} - 2q_{1,i} + q_{0,i}| < \beta/8 \quad i=0\ (2),\ i=3\ (3)$$

$$|p_{3,i} - p_{0,i}| + |q_{0,i} - q_{3,i}| < \beta/8 \quad i=0\ (4),\ i=3\ (5)$$

$$|p_{0,i} - q_{0,i}| < 2.5t_c \quad i=0\ (6),\ i=3\ (7)$$

$$|p_{2,0} - 2p_{1,0} + p_{0,0}| + |p_{2,3} - 2p_{1,3} + p_{0,3}| < 3/16\beta \quad (8)$$

$$|q_{2,0} - 2q_{1,0} + q_{0,0}| + |q_{2,3} - 2q_{1,3} + q_{0,3}| < 3/16\beta \quad (9)$$

$$|\delta_0| < 10tc \quad i=0..3 \quad (10)$$

After computing the filtering decisions, this unit computes the filtering operations for normal and strong filters . Normal filter modifies 1 or 2 samples along the block boundary. It computes the delta values (i.e. delta0, deltap1, deltap2) which are offsets that must be added to the original samples (p0, p1, q0 and q1) in order to generate the final filtered sample (p0', p1', q0' and q1'). They are applied to the four rows of samples of 4x4 blocks P and Q. Delta 0 operation is always computed when the normal filter is selected in filtering decision process and modifies the p0 and q0 samples that are close to the boundary. Delta p1 and delta q1 operations modify also p1 and q1 samples. In the design of this unit we have also replaced the multiplications by a sequence of adders and shift operations.

Strong filter always modify 3 samples along the block boundary. It computes the delta values (i.e. delta0s, delta1s, delta2s) which are offsets that must be added to the original samples (p0, p1, p2, q0, q1 and q2) in order to generate the final filtered sample (p0', p1', p2', q0', q1' and q2'). We applied to the four rows of samples of 4x4 blocks P and Q. Similar to normal filter data paths, we have also replaced the multiplications by adders and shift operations.

4. Experimental setup

The proposed method implemented in Xilinx 13.2 using Verilog code, which gives the FPGA results such as power consumption, slice LUT, slice registers, and delay. Modelsim SE 10.1 software is used to verify the Verilog timing diagram. The entire work is done by using I_7 system with 8 GB RAM.

V. Result and discussion

Performance measured	**Existing system**	**Proposed system**
Device	Virtex 6 FPGA	Virtex 6 FPGA
Power consumption(W)	2.781	2.654
Slice LUTs	2147	2112
Slice register(memory)	488	462
Delay(ns)	6.958	6.842

Tab.1. Comparison of Power, LUT, slices and delay value for existing and proposed method

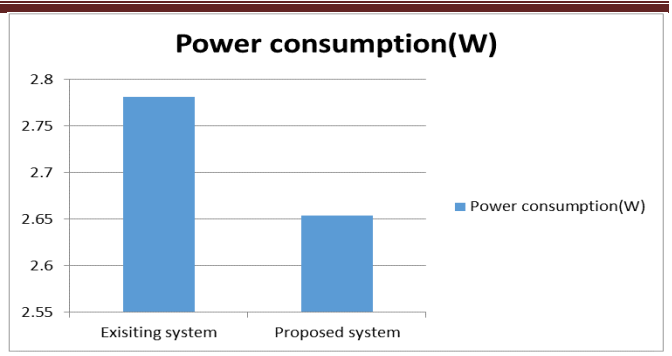

Fig.3 Comparison of power (W) with proposed and existing system

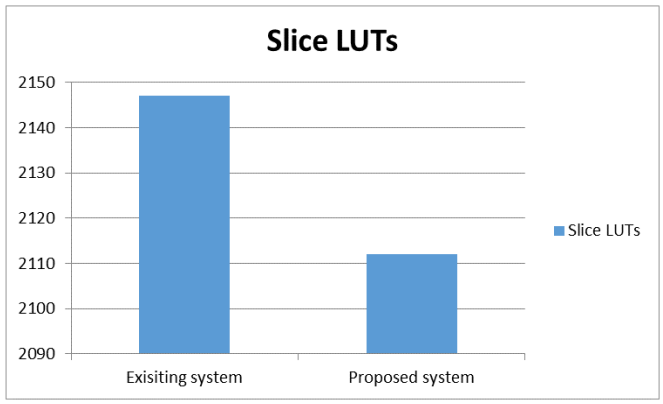

Fig.4 Comparison of slice LUTs with proposed and existing system

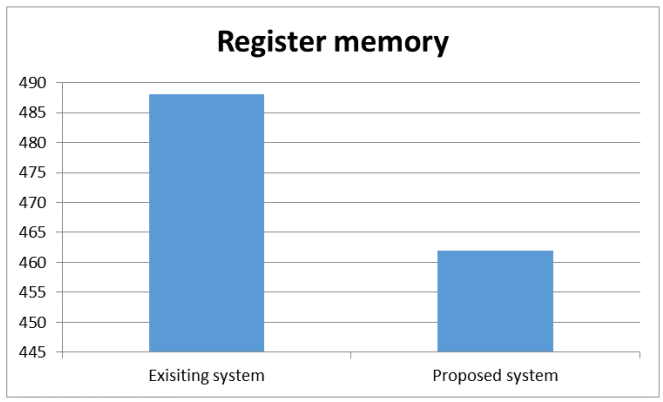

Fig.5 Comparison of register memory with existing and proposed system

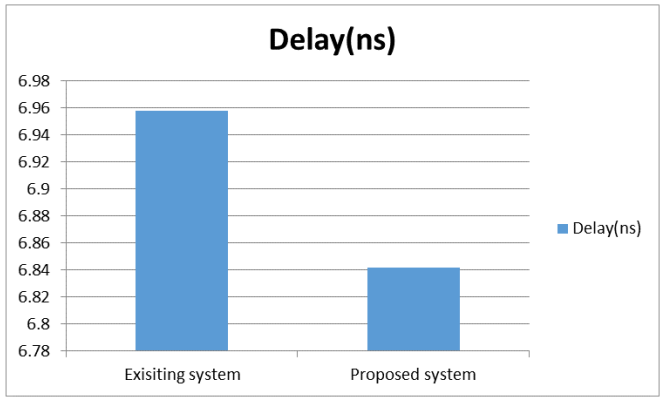

Fig.6 comparison of Delay (ns) with existing and proposed method

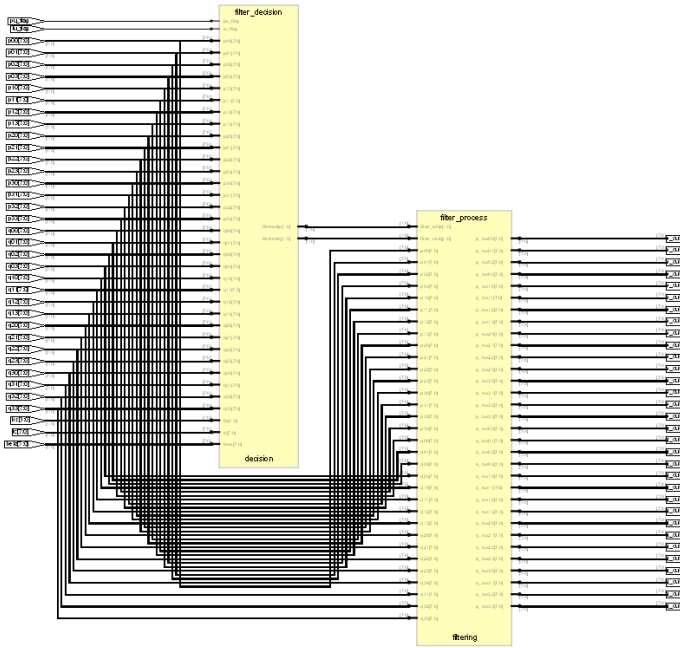

Fig.7. RTL schematic of filter architecture

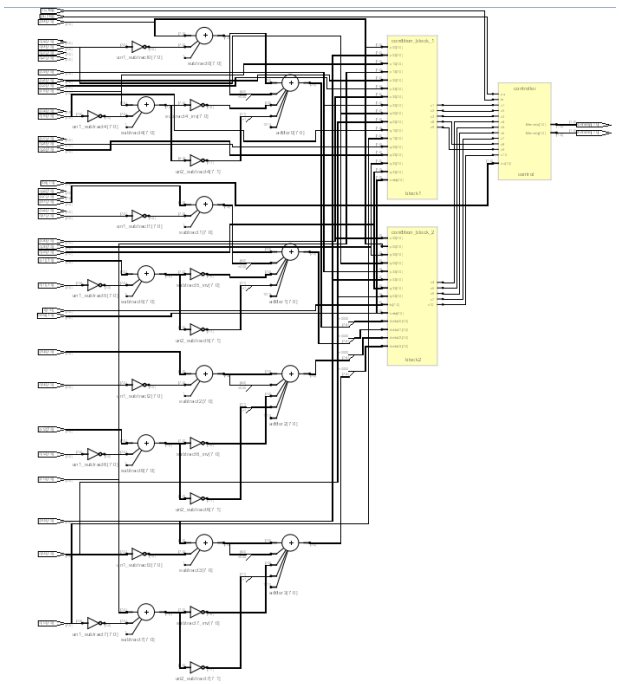

Fig.8 RTL schematic of filter decision

VI. Conclusion

The H.264/AVC video coding standard introduces as a novelty with respect to its predecessors andeblocking filter allocated at the reconstruction loop of the encoder and the decoder. This filter greatly contributes to provide a better visual quality for the user and to enhance the rate-distortion performance of H.264/AVC-based video coding systems. However, due to its highly adaptive and computationally intensive nature, it could seriously jeopardize the real-time performance of the target video application, especially for HDTV systems. Due to this reason, efficient VLSI deblocking filter architectures are a must in the majority of nowadays video coding commercial products.

References

[1] Yahya, Ali Abdullah, Jieqing Tan, and Lian Li. "Video noise reduction method using adaptive spatial-temporal filtering." *Discrete Dynamics in Nature and Society* 2015 (2015).

[2] Prof. Meenakshi Sharma, Mukhvinder Singh *et la "*Adaptive Threshold Based Video Compression Technique" Volume 4, Issue 7-916

[3] Sahni, Pearl, and Gaganjot Kaur. "Study of Various Filtering Techniques, Genetic Algorithm and Fuzzy Logic for Reducing Blocking Artifacts in Compressed Images."

[4] Silpa, K., and S. Aruna Mastani. "NEW APPROACH OF ESTIMATING PSNR-B FOR DEBLOCKED IMAGES." *International Journal of Advances in Engineering & Technology* 7.1 (2014): 183.

[5] K. Rajalakshmi, K. Mahesh, PhD *et la "*A Review on Video Compression and Embedding Techniques*"* Volume 141 – No.12.

[6] Abhilasha Agarwal, Anurag Bhatt *et la* "Study, Design and Deployment of a Video Compression Technique through Crossbreeding Discrete Wavelet Transformation & Discrete Cosine Transformation" Volume 5, Issue 3-858.

[7] M.Arriukannamma, J.G.R.Sathiaseelan *et la "*A Lossy Adaptive Multiwavelet Transform for High Quality Compression Ratio Using Medical Endoscopic Video" Vol.5 Issue.4 pg. 50-61.

[8] Yim, Changhoon, and Alan Conrad Bovik. "Quality assessment of deblocked images." *IEEE Transactions on Image Processing* 20.1 (2011): 88-98.

[9] Sunkara, Jaya Krishna, et al. "A New Video Compression Method using DCT/DWT and SPIHT based on Accordion Representation." *International journal of Image, Graphics and Signal Processing* 4.4 (2012): 28.

[10] Shaikh, Muhammad Aakif, and Sagar S. Badnerkar. "Video Compression Algorithm Using Motion Compensation Technique: A Survey." *International Journal* 2.3 (2014).

[11] Norkin, A., Bjontegaard, G., Fuldseth, A., Narroschke, M., Ikeda, M., Andersson, K., Zhou, M. and Van der Auwera, G., 2012. HEVC deblocking filter. IEEE Transactions on Circuits and Systems for Video Technology, 22(12), pp.1746-1754.

[12] Esche, M., Glantz, A., Krutz, A. and Sikora, T., 2012. Adaptive temporal trajectory filtering for video compression. IEEE Transactions on Circuits and Systems for Video Technology, 22(5), pp.659-670.

[13] C. Karthikeyan and Dr. Rangacha, "IMPLEMENTATION OF DEBLOCKING FILTER ALGORITHM USING RECONFIGURABLE ARCHITECTURE," International Journal of Application or Innovation in Engineering & Management (IJAIEM), Volume 2, Issue 12, December 2013.

[14] Ozcan, E., Adibelli, Y. and Hamzaoglu, I., 2013. A high performance deblocking filter hardware for high efficiency video coding. IEEE Transactions on Consumer Electronics, 59(3), pp.714-720.

[15] Vanne, J., Viitanen, M., Hamalainen, T.D. and Hallapuro, A., 2012. Comparative rate-distortion-complexity

analysis of HEVC and AVC video codecs. IEEE Transactions on Circuits and Systems for Video Technology, 22(12), pp.1885-1898.