The Memory efficient Noval Modified Deblocking Filter order for HEVC Systems

Mr.M.Vasanthakumar,vassece.win@gmail.com, Assistant Professor, AVS Engineering College,Salem, Tamilnadu.
Mr.G.Kanagaraj,(raj.techstorm@gmail.com), Assistant Professor, AVS Engineering College,Salem, Tamilnadu.
M.Nandhini,(mnandhiniece15@gmail.com) PG Scholar,AVS Engineering College, ,Salem, Tamilnadu.

ABSTRACT

The new deblocking filter (DF) tool of the next generation High Efficiency Video Coding (HEVC) standard is one of the most time consuming algorithms in video decoding. In order to achieve real-time performance at low-power consumption, we developed a hardware accelerator for this filter. This paper proposes high throughput hardware architecture for HEVC deblocking filter employing hardware reuse to accelerate filtering decision units with a low area cost. Our architecture achieves either higher or equivalent throughput with 5X-6X lower area compared to state of-the-art deblocking filter architectures. The proposed architecture of this paper analysis the logic size, area and power consumption using Xilinx 14.2.

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 survey

"A Deblocking Filter Hardware Architecture for the High Efficiency Video Coding Standard," The new deblocking filter (DF) tool of the next generation High Efficiency Video Coding (HEVC) standard is one of the most time consuming algorithms in video decoding. In order to achieve real-time performance at low-power consumption, we developed a hardware accelerator for this filter. This paper proposes high throughput hardware architecture for HEVC deblocking filter employing hardware reuse to accelerate filtering decision units with a low area cost. Our architecture achieves either higher or equivalent throughput (4096x2048 @ 60 fps) with 5X-6X lower area compared to stateof-the-art deblocking filter architectures. "IMPROVED SAMPLE ADAPTIVE OFFSET FOR HEVC," Sample adaptive offset (SAO) is the new in-loop filter in High Efficiency Video Coding (HEVC) standard. In this paper, the problemsin the early version of SAO technique is discussed, and it is shownhow the proposed methods improve its performance. It is proposedto restrict edge offset sign according to edge shape to reduce visualartifact caused by edge offset. It is also proposed to reduce number of band offset to facilitate implementation. Experimental resultsshows the proposed methods effectively reduce the artifact causedby edge offset, and reduce number of band offset by half withoutcompromising coding efficiency. "Sample Adaptive Offset in the HEVC Standard," This paper provides a technical overview of a newlyadded in-loop filtering technique, sample adaptive offset (SAO),in High Efficiency Video Coding (HEVC). The key idea of SAOis to reduce sample distortion by first classifying reconstructedsamples into different categories, obtaining an offset for eachcategory, and then adding the offset to each sample of thecategory. The offset of each category is properly calculated atthe encoder and explicitly signaled to the decoder for reducingsample distortion effectively, while the classification of eachsample is performed at both the encoder and the decoder forsaving side information significantly. To achieve low latency ofonly one coding tree unit (CTU), a CTU-based syntax design isspecified to adapt SAO parameters for each CTU. A CTU-basedoptimization algorithm can be used to derive SAO parameters ofeach CTU, and the SAO parameters of the CTU are inter leavedinto the slice data. It is reported that SAO achieves on average3.5% BD-rate reduction and up to 23.5% BD-rate reduction withless than 1% encoding time increase and about 2.5% decodingtime increase under common test conditions of HEVC referencesoftware version 8.0.

III Existing System

The new High Efficient Video Coding (HEVC) standard provides approximately 50% bit-rate reduction compared to the last version of H.264/AVC (Advanced Video Coding) standard with similar subjective video quality.

To achieve such compression efficiency, especially focusing on high and ultrahigh resolution videos (beyond HD

1920x1080 pixels), HEVC employs larger block sizes (up to 64x64 pixels) for prediction. A new quadtree structure splits those blocks hierarchically down to 4x4-pixel elementary blocks. A large set of new advanced coding tools was introduced in HEVC. The higher compression efficiency achieved with HEVC results in an increase of the computational complexity when compared to H.264/AVC. The new Deblocking Filter (DF) is included in both HEVC video encoder and decoder to reduce the blocking artifacts present in a reconstructed video, which are introduced by the inherent block partitioning and strong quantization in video encoding. DF contributes to up to 6% bit-rate reduction (1.3%-3.3% bit-rate reduction on average) for the same video quality. Although DF is an optional feature invideo encoder (and thus may not be used in video decoder), it is often employed because of the high bit-rate reduction. DF is one of the most time consuming tools of HEVC video decoder. We have profiled the HEVC reference decoder software (HM 10.0) to identify how much of the execution time belongs to the DF process. The profiling was performed by decoding 3 seconds of four ultrahigh resolutions (with 2560x1600 pixels) and five high resolution (with 1920x1080 pixels) encoded video sequences. Encoded video sequences were generated with the followingencoder configuration: (i)Random Access (RA) configuration1with Group of Pictures (GOP) equal to 8 (ii)Intra period2for each video sequence is defined as depending upon the specific frame rate of the video sequence, e.g. 24, 30, 50 or 60 frames per second (fps); (iii) each sequence is encoded with four different Quantization Parameter (QP) values QP={22,27,32,37}as defined in the HEVC Common Test Conditions.

IV Proposed System

We use the image of size 64x64 pixels, so we split the image into 32x32,16x16,8x8 and 4x4 pixels then do the de-blocking filter.

The process flow of the de-blocking is shown in fig 1.

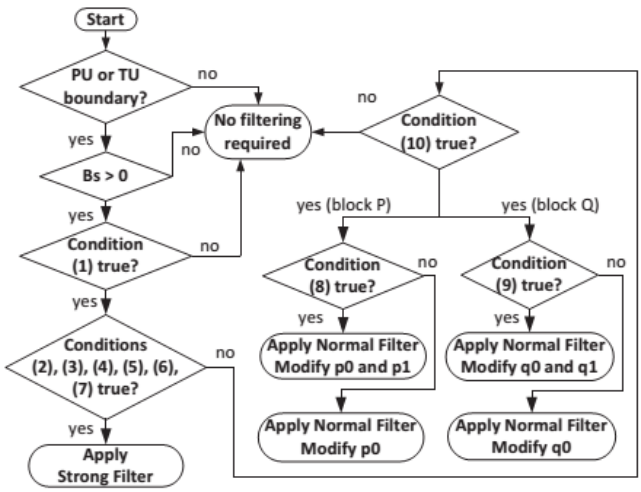
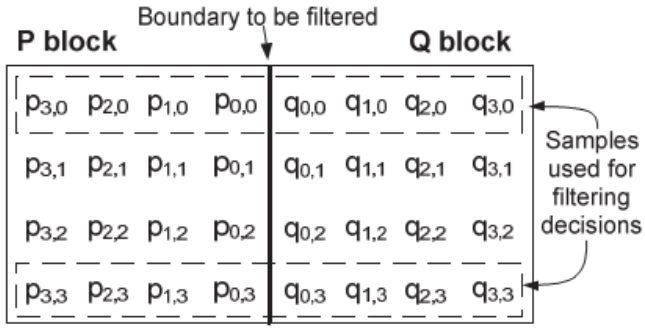


Figure 1: De-blocking filter flow

The deblocking filter reduces the blocking artifacts (visible discontinuities in the vidEO) caused by block-based encoding with strong quantization. It is applied by modifying samples along horizontal and vertical boundaries of PUs and TUs of size not smaller than 8x8 samples. Filtering is applied separately in 4x4 blocks (so-called P and Q blocks), as shown in Fig. 1.Normal and strong filtering modes modify 2 and 3

luma samples along each boundary, respectively. The example in Fig. 2 shows a vertical boundary, as the horizontal boundary filtering is analogous



Filtering decisions (conditions):

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

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

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

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

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

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

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

Figure 3: filtering decisions (conditions).

Our design has two main units:

(1) Filtering decisions

(2) Filtering operations

Filtering decisions:

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 merged datapathfor 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. With that, those five conditions are generated in only one clock cycle.

Datapaths for the remaining conditions are depicted in Fig. 6. Datapaths for conditions (4) and (5) are equal, only differing by the input samples (condition 4 for the first row and condition 5 for the last row of 4x4 blocks). We have included two instances of this datapath to compute both conditions (c4 and c5) in the same clock cycle. The same was made for conditions (6) and (7). Condition (10) is an additional filtering decision applied to ¡0for the four rows of 4x4 blocks after normal filtering operation. Our architecture includes two instances of this datapath in the design to compute c10 for the four rows in two clock cycles. Each instance

computes two rows of samples. Additional datapaths for β and tc multiplications, needed to compute all the conditions,

are also shown in Fig. 6. β and tc values are generated by a lookup table with QP value as input index

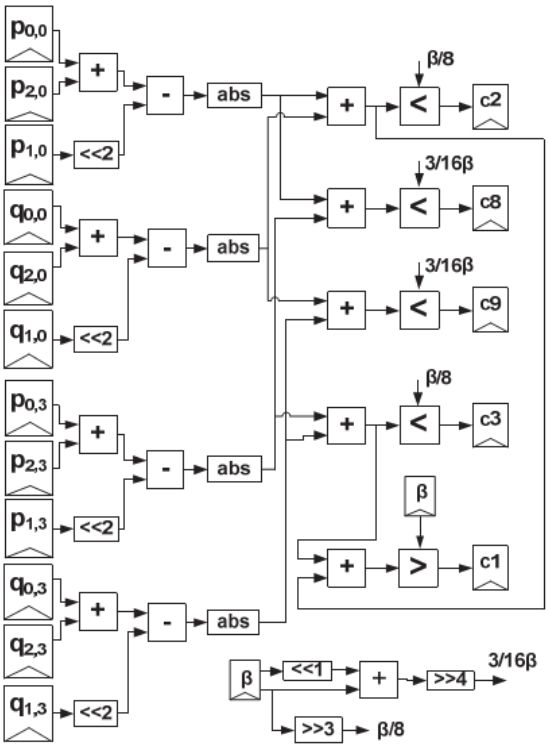


Figure 3: Merged datapath for conditions (1), (2), (3), (8) and (9).

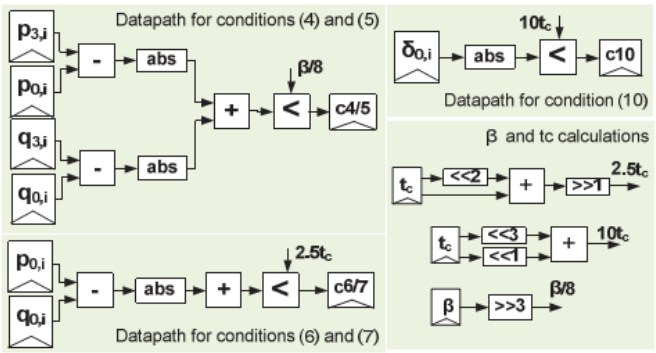
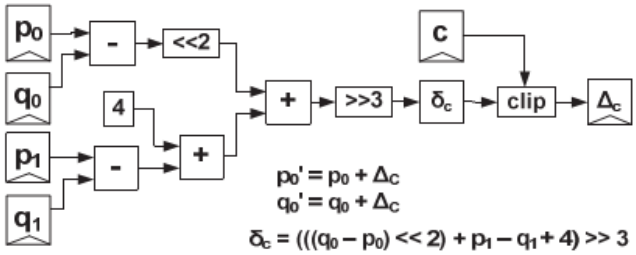


Figure 4: Datapaths for conditions (4), (5), (6), (7) and (10).

Chroma filter:



In the design of this unit we have also replaced the multiplications by a sequence of adders and shift operations. Our architectural design includes two instances of each datapath shown in Fig. 8

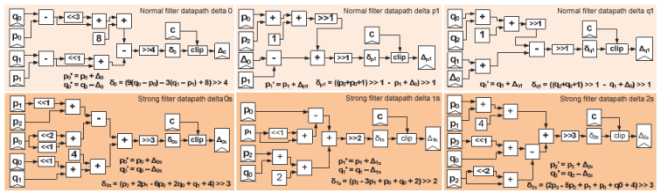


Figure 5: Datapaths for normal and strong filtering operations.

Boundary strength and Edge level adaptivity

Boundary strength(BS) is calculated for boundaries that are either prediction unit boundaries or the transform unit boundaries.The boundary strength can take values as 0,1 or 2.For luma component, only block boundaries with BS values 1 or 2 are filtered. Therefore there is no filtering in static areas which avoids multiple subsequent filtering of same areas where pixels are copied from one picture to another with a residual equal to zero which can cause over smoothening.In case of chroma components, only boundaries with BS equal to 2 are filtered. Hence the block boundaries that are filtered have at least one of the two adjacent blocks intra predicted.

Secured communication mainly base on cryptography, which encrypts plain text to generate cipher text. However, the transmission of cipher text may easily arouse attackers" suspicion, and the cipher text may thus be intercepted, attacked or decrypted violently. In order to make up for the shortcomings of cryptographic techniques, steganography has been developed as a new covert communication means in recent years. It transfers message secretly by embedding it into a cover medium with the use of information hidingtechniques.

V Results

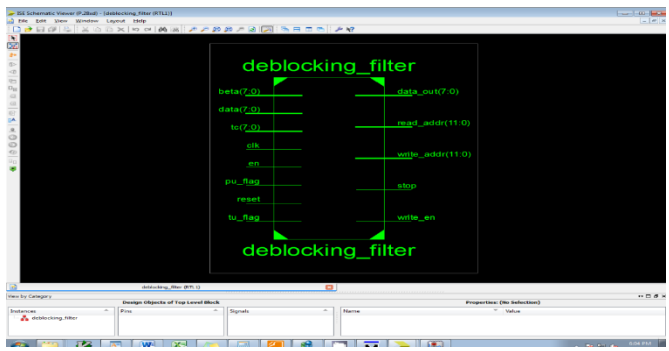


Fig 6: RTL top View for DF

Table 1: Comparison Table for Existing and Proposed system

	Exisiting system	**Proposed system**
Device	Vertix 6 FPGA	Vertix 6 FPGA
Power consumption(W)	2.707 w	2.656

Slice LUTs	2285	2121
Slice register(memory)	508	469
Delay(ns)	6.956	6.842

VI . Conclusion

A low complexity Deblocking filter order is proposed to improve video frame quality . With the help of proposed method, the complexity of deblocking filter is reduced by comparing synthesis results of virtx6 fpga. By this deblocking filter order register memory, Look up table ,slices are reduced when compare to existing system.

References

[1]. J. Zhu, D. Zhou, G. He, and S. Goto, "A combined SAO and de-blockingfilter architecture for HEVC video decoder," inProc. IEEE 20th Int. Conf.Image Process., Sep. 2013, pp. 1967–1971.

[2]. Description of Core Experiment 1 (CE1): Sample Adaptive Offset Filtering, JCTVC-H1101, eb. 2012.

[3]. R. Song, H. Cui, Y. Li, and X. Song, "A five-stage pipeline design ofbinary arithmetic encoder in H.264/AVC," inProc. Asia-Pacific SignalInform. Process. Assoc. Annu. Summit Conf., Dec. 2012, pp. 1–4.

[4]. V. Rosa, L. Max, and S. Bampi, "High performance architectures for thearithmetic encoder of the H.264/AVC CABAC entropy coder," inProc.IEEE 17th Int. Conf. Electron., Circuits, Syst., Dec. 2010, pp. 383–386.

[5]. S. M. Kay, Fundamentals of Statistical Signal Processing, Volume I:Estimation Theory, 1st ed. Englewood Cliffs, NJ, USA: Prentice-Hall,1993.

[6]. Calculation of Average PSNR Differences Between RD Curves,ITU-TSC16/Q6, VCEG-M33, 2001.

[7]. HM Reference Software. (2013). [Online]. Available: svn://hevc.kw.bbc.co.uk/svn/jctvc-hm/tags/ HM-9.0. Accessed on: Jan. 2014.

[8]. C.-A. Chien, H.-C. Chang, and J.-I. Guo, "A high throughput in-loop deblocking filter upporting H.264/AVC BP/MP/HP video coding," inProc.

[9]. Description of Core Experiment 1 (CE1): Sample Adaptive Offset Filtering, JCTVC-H1101, eb. 2012

[10]. J. Zhu, D. Zhou, G. He, and S. Goto, "A combined SAO and de-blockingfilter architecture for HEVC video decoder," inProc. IEEE 20th Int. Conf.Image Process., Sep. 2013, pp. 1967–1971.

[11]. IEEE Asia Pacific Conf. Circuits Syst., Nov.-Dec. 2008, pp. 312–315.

[12]. K. Xu and C.-S. Choy, "A five-stage pipeline, 204 Cycles/MB, singleport SRAM-based deblocking filter for H.264/AVC,"IEEE Trans. CircuitsSyst. Video Technol., vol. 18, no. 3, pp. 363–374, Mar. 2008.

[13]. Y.-C. Lin and Y.-L. Lin, "A two-result-per-cycle deblocking filter architecture for QFHD H.264/AVC decoder," IEEE Trans. Very Large ScaleIntegr. Syst., vol. 17, no. 6, pp. 838–843, Jun. 2009.

[14]. T.-M. Liu, W.-P. Lee, and C.-Y. Lee, "An in/post-loop deblocking filterwith hybrid filtering schedule,"IEEE Trans. Circuits Syst. Video Technol.,vol. 17, no. 7, pp. 937–943, Jul. 2007.

[15]. N. T. Ta, J. S. Youn, H. G. Kim, J. R. Choi, and S.-S. Han, "Low-powerhigh-throughput deblocking filter architecture for H.264/AVC," inProc.Int. Conf. Electron. Comput. Technol., Feb. 2009, pp. 627–631.

[16]. M. Nadeem, S. Wong, G. Kuzmanov, and A. Shabbir, "A high-throughput,area-efficient hardware accelerator for adaptive deblocking filter inH.264/AVC," inProc. IEEE/ACM/IFIP 7th Workshop Embedded Syst.Real-Time Multimedia, Oct. 2009, pp. 18–27.