Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2015, 2(5): 38-43

Research Article ISSN: 2394 - 658X

Parallel Architecture for De-blocking Filter in High Efficiency Video Coding (HEVC)

Nguyen Thanh Loi, Nguyen Nam Hai, Ngo Vu Duc, Nguyen Hoang Dung and Thai Nam Son

School of Electronics and Telecommunication, Hanoi University of Science and Technology, Hanoi, Vietnam
dung.nguyenhoang@hust.edu.vn

ABSTRACT

With the desire to improve performance of the HEVC's De-blocking Filter, a new filtering order is proposed. This method maximally supports for parallel processing by dissolving the data dependency between the adjacent filtering operations. This order allows handling the horizontal edges and vertical edges at the same time. In this paper, we implement the HEVC De-blocking Filter architecture based on the new filtering order in a Xilinx Virtex 7 FPGA. This design has maximum clock frequency 113MHz, provide throughput up to 38 Gbps.

Key words: Video compression, H265/HEVC, De-blocking Filter, Parallel processing, Field Programmable Gate Array

INTRODUCTION

Join collaborative team on video coding (JCT-VC) recently developed a new international video compression standard called High efficiently video coding (HEVC or H265) [1]. Comparison with the H264/AVC high profile, the HEVC standard is expected that halved bit rate means that the compression performance is doubled. To achieve that, the H265 standard uses a lot of difference tools; one of these tools is the new De-blocking filter algorithm.

In the HEVC standard, quad-tree based coding structure is an important feature, there are several new coding structure is applied such as: coding tree unit (CTU), coding unit (CU), prediction unit (PU) and transform unit (TU). CU is the basic unit for inter/intra coding, CU size can change from 8x8 pixels to 64x64 pixels. Coupled with CU, PU contains the information which is used for prediction processing and TU is used for transform as well as quantization operation [2]. Same as the previous video compression standards, the HEVC divides a frame into CUs and perform some operations with each CU separately. This causes correlation loss between CUs and discontinuities on the edges of CUs. Therefore, reconstructed frames suffer from blocking artifacts. The De-blocking Filter is employed to improve the subjection and objection quality of the video by reducing the blocking artifacts occurring from the quantization of transform coefficients and the block-based motion compensation. HEVC also uses in-loop De-blocking filter similar to H264/AVC. As mention above, CU, TU and PU is three basic units in HEVC, and boundaries of these unit is involved in the De-blocking filter process. All filter operations are applied to 8x8 pixels block boundaries, not 4x4 pixels block as H264/AVC. An extreme important for HEVC De-blocking filter is to decide whether to filter or not, if the filtering is decided how is strength of the filtering? The detailed algorithm of the De-blocking filter is presented in the H.265/HEVC standard specification [3].

Almost architecture of De-blocking filter implement traditional filtering order, it is edge-by-edge, the vertical edges is filtered first from left to right, following is the horizontal edges from top to bottom. Some architecture try to break this sequential order but degree of parallelism is not high. However, with the new proposed filtering order, ours architecture can simultaneously performs both horizontal and vertical filter operations. So degree of parallelism is also increased. We focus on improvement of performance, the trade-off is hardware cost but this not important in HEVC design.

RELATED WORK

In this paper, we consider the case of a 32x32 pixels luminance CTU and two 16x16 pixels chrominance CTU for illustration. Fig. 1 shows the horizontal edges and the vertical edges between adjacent 8x8 pixels CUs of the 32x32 pixels CTU.

The filtering operation in HEVC is always done from left to right and from top to bottom. Chrominance CUs are filtered in a similar order of the luminance CUs. According to the above rule, traditional filtering order is proposed in Fig. 2. There are 48 edges include both luminance and chrominance CUs, filtered in this order. The filtering operation start at the first vertical edge which is between two 4 two 8x8 luminance CU L1 and C1. Similarly, filtering operation is performed with the vertical edges from 2 to 24 and the horizontal edges from 25 to 48. Because the filtered pixels on which the vertical filter operation is performed are used for the horizontal filtering later, these values can be stored in the on-chip SRAM. Therefore, size of on-chip SRAM and access time increase significantly.

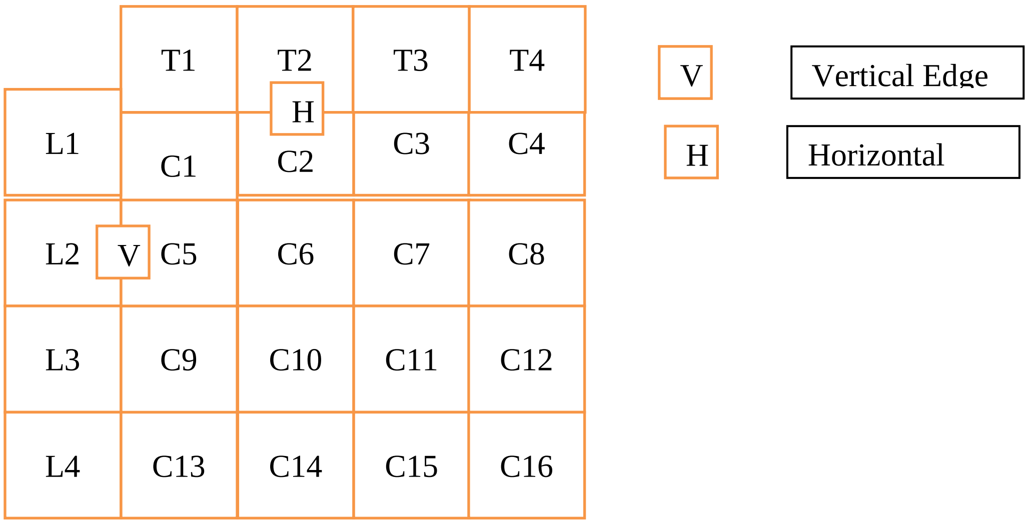

Fig. 1 Horizontal and vertical edges of 32x32 luminance CTU

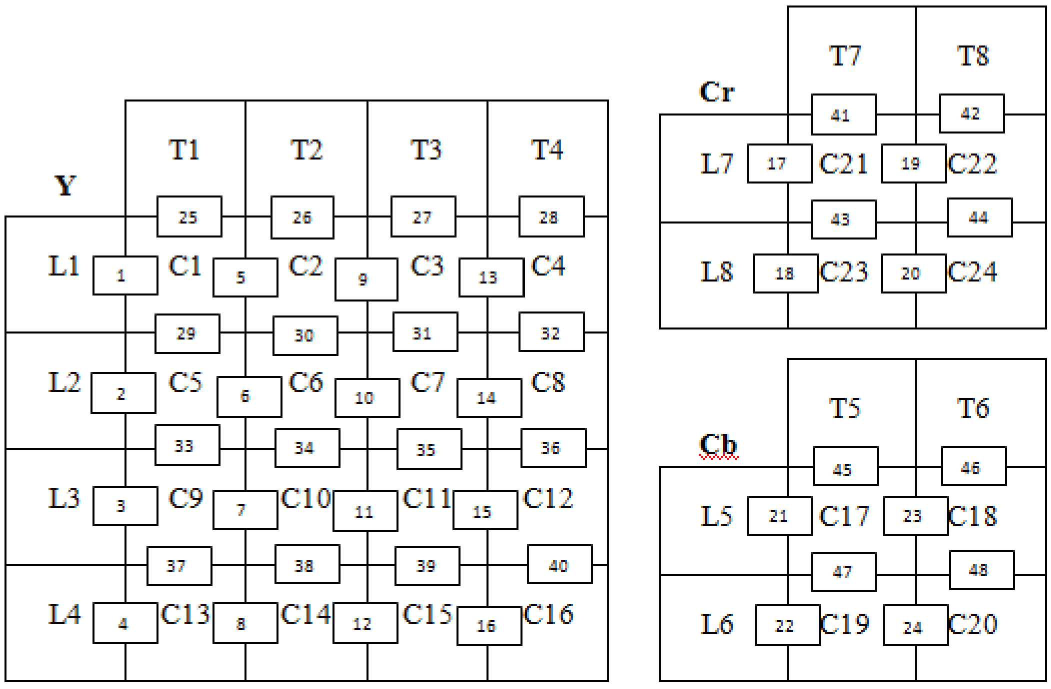

Fig. 2 Traditional HEVC filtering order

Based on traditional filtering order, there are some difference HEVC De-blocking filter architecture have been presented. Shen [4] showed a pipelined architecture and a new memory organization, which help reduce access time of SRAM and make simply filtering operation because it is very close with H264 filter processing. But this method is basically sequential; degree of parallelism is very low. Dalcin [5] used a creative datapath in his architecture to perform filtering operation, but the implementation method is not optimal and the detail of memory design is not clearly provided. Huong [6] proposed a new and innovative filtering order, this order maximally supports for parallel processing. However her design can only handles 8 pixels per cycle and the way of organizing memory not optimal, so the throughput not as high as expected. After researching all architecture above, we decided to choose Huong 's filtering order and propose a new parallel De-blocking filter architecture, which employs that order, to overcome the disadvantages. The detail of them will be explained in next section.

THE PROPOSED ARCHITECTURE

Parallel Filtering Order

Fig. 3 shows the new parallel processing order, which is proposed in [6]. With this order, the HEVC De-blocking filter simultaneously executes two parallel horizontal filtering and two parallel vertical filtering. In the first step, vertical edges between pair of CUs (L1-C1) and (C1-C2) are filtered almost at the same time. Then the horizontal filter operations between CUs C2 and C3 are performed in second step. The horizontal filter operations between CUs C3 and C4 are also performed simultaneously. This process is performed similarly for the next rows. From the 4th step, when two vertical edges between (C6-C7) and (C7-C8) is being filtered, the vertical filter operations between (T1-C1) and (C1-C5) are performed simultaneously. The processes of the next columns are similar to that of the first column. The process of each data flow in two chrominance CUs is similar to that in the luminance CU. According to this order, after eleven steps the De-blocking filter of the CTU 32x32 pixels can be finished.

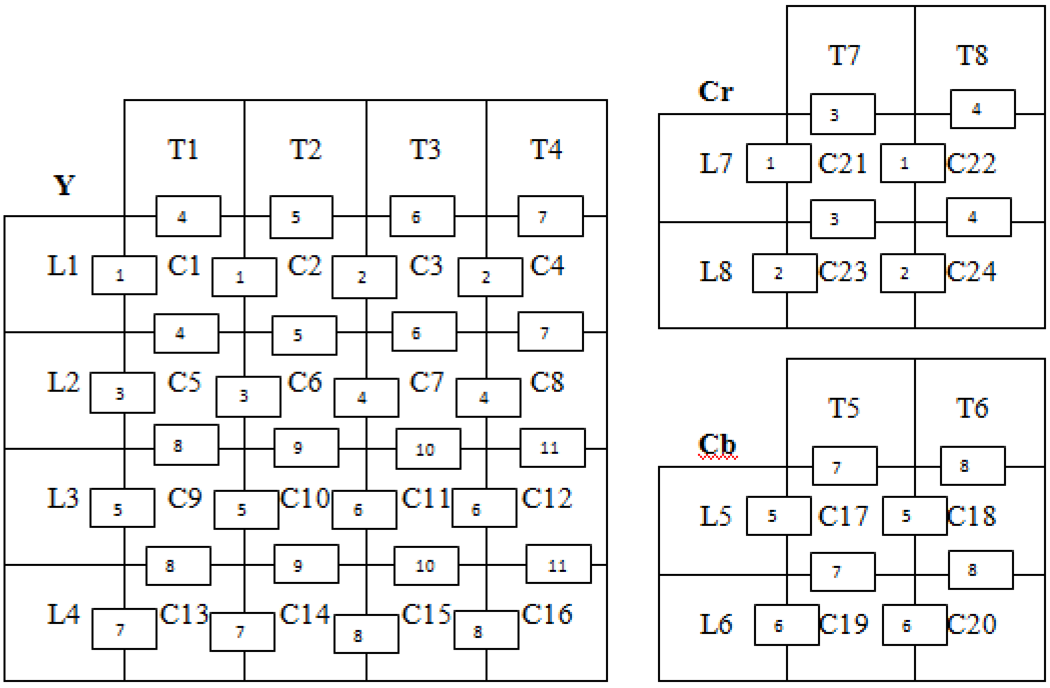

Fig. 3 The new parallel filtering order

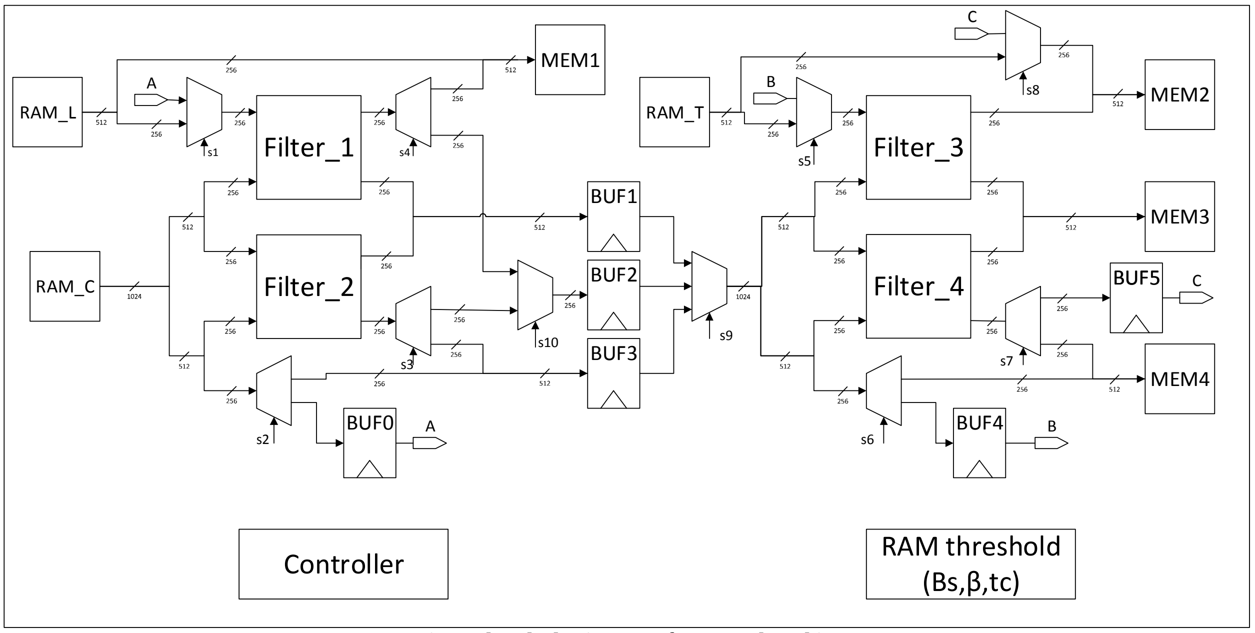

Fig. 4 The Block Diagram of Proposed Architecture

Parallel De-Blocking Filter Architecture

In this section, a parallel De-blocking filter architecture employing the new processing order above is introduced. Fig. 4 shows the proposed architecture. There are five main components in this architecture. The first component is CUs memory organization, these are Dual-port SRAMs to store pixel value and threshold value, includes : two

8x512 bits dual-port SRAM (RAM_L and RAM_T) store the left neighbour CUs (from L1 to L8) and the top neighbour CUs (from T1 to T8) respectively, one 20x512 bit dual-port SRAM store the current CUs (from C1 to C20) called RAM_C and the last is a SRAM store threshold value, which is put into filter unit. The second component is filter module; include blocks named Filter1, Filter2, Filter3, and Filter4. Module Filter1 and Filter2 handle horizontal filtering, module Filter3 and Filter4 handle vertical filtering. Each block is data paths with the input are pixel of CUs and threshold, the output are filtered pixel. The detail of module filter will be explained in the following subsection. The third component is module Controller; it generates all of control signals and enables signals for the whole architecture. The fourth component is set of Buffer (includes BUF1, BUF2, BUF3, BUF4, BUF5), in fact each buffer is a Dual-port SRAM, it used to store temporary value after horizontal filtering and before vertical filtering. The last component is output memory (includes MEM1, MEM2, MEM3, MEM4), they are simple RAMs used to store filtered pixel of CTU.

Table -1 Data Flow of the Proposed Architecture

Step / Filter	1	2	3	4	5	6	7	8	9	10	11
Filter_1	½ L1	½ C2	L2	½ C6	L3	½ C10	L4	½ C14			
Filter_1	½ C1	½ C3	½ C5	½ C7	½ C9	½ C11	½ C13	½ C15			
Filter_2	½ C1	½ C3	½ C5	½ C7	½ C9	½ C11	½ C13	½ C15			
Filter_2	½ C2	½ C4	½ C6	½ C8	½ C10	½ C12	½ C14	½ C16			
Filter_3				T1	T2	T3	T4	½ C5	½ C6	½ C7	½ C8
Filter_3				½ C1	½ C2	½ C3	½ C4	½ C9	½ C10	½ C11	½ C12
Filter_4				½ C1	½ C2	½ C3	½ C4	½ C9	½ C10	½ C11	½ C12
Filter_4				½ C5	½ C6	½ C7	½ C8	½ C13	½ C14	½ C15	½ C16

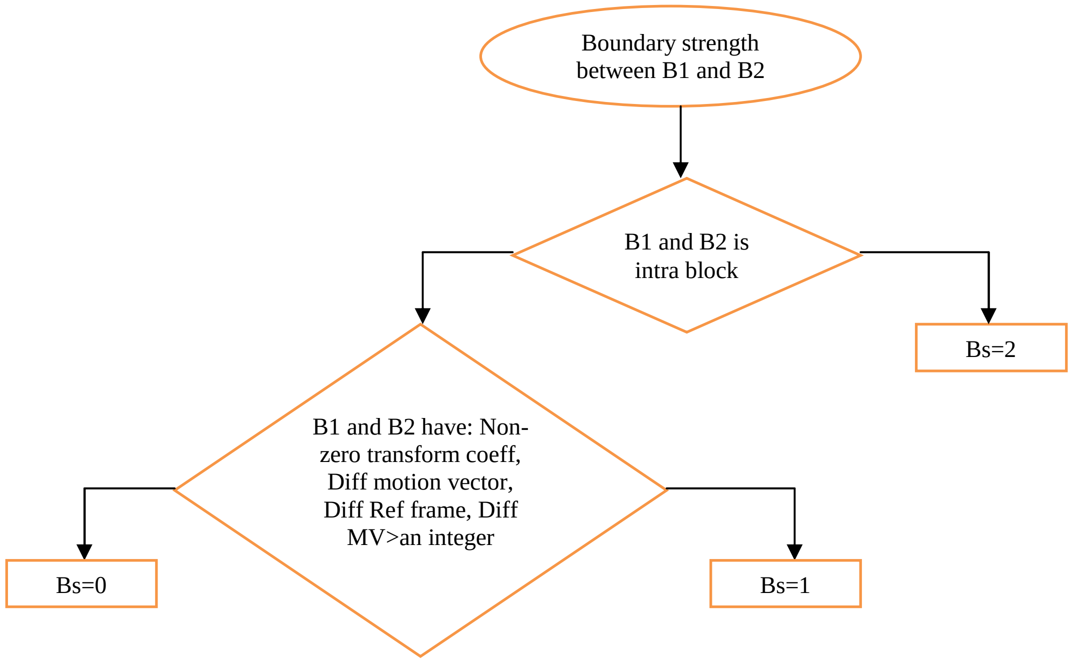
Boundary strength between B1 and B2
B1 and B2 is intra block
Bs=2
B1 and B2 have: Non-zero transform coeff, Diff motion vector, Diff Ref frame, Diff MV>an integer
Bs=0
Bs=1

Fig. 5 Flowchart of the boundary strength determination

Step 1, RAM_L reads out one CU is L1 (512 bit) and RAM_C reads out two CUs are C1, C2 (1024 bit). Then a haft of L1 and a haft of C1 are taken to module Filter1, in this module vertical edge between L1 and C1 is filtered. At the same time, module Filter2 receives the remaining haft of C1 and a haft of C2, so the edge between C1 and C2 is filtered. After filtering, a haft of L1 and a haft of C2, which is not filtered, will be stored in MEM1 and BUF0 respectively, C1 is combined and stored in BUF1, a haft filtered of C2 is stored in BUF2.

Step 2, RAM_C reads out two CUs C3, C4. Then a haft of C2 from BUF0 is transferred to Filter1 together with a haft of C3. Simultaneously, a remaining haft of C3 and a haft of C4 are transferred to Filter2, so in this step edge between C2-C3 and C3-C4 is filtered. After that, part of C2 is filtered will be stored in BUF2, total C3 is combined and stored in BUF1,a haft of C4 not filtered and a haft of C4 from output of module Filter2 will be mixed and stored in BUF3. Finish step 1 and step 2, we handles first row of CTU. With the next rows, we do similarly.

In step 4, while we apply horizontal filtering for edges between C6-C7 and C7-C8, BUF1 reads out two filtered CUs C1, C5 and they are transposed from row to column. After that, a haft of T1 from RAM_T and a haft C1 from BUF1 are transferred to Filter3. Filter4 receives the remaining haft of C1 from BUF1 and a haft of C5 from BUF1 at the same time. So when finishing step 4, horizontal edges between T1-C1 and C1-C5 are filtered. Next steps, we performs similar step 4.

Finally, after we filter all edge, the filtered pixel will be store in output memory (C1, C2, C3, C4, C9, C10, C11, C12 are stored in MEM3, C5, C6, C7, C8 are stored in MEM2, C13, C14, C15, C16 are stored in MEM4). To explain the proposed parallel De-blocking filter architecture more easily, the data flow is shown in Table 1.

Boundary Strength (Bs)

The strength of De-blocking filter in H.265/HEVC is controlled by the value of several syntax elements similar to that of H.264/AVC except that only three strengths from 0 to 2 are used rather than five. There are two filter modes: strong and weak for the De-blocking filter. For the luminance component, only block boundaries with Bs values equal to 1 or 2 are filtered. The filter modes are defined and selected based on the Bs value. Fig. 5 shows the flowchart of the boundary strength determination.

Datapath for Module Filter

Fig. 6 shows the architecture of filter's datapath.

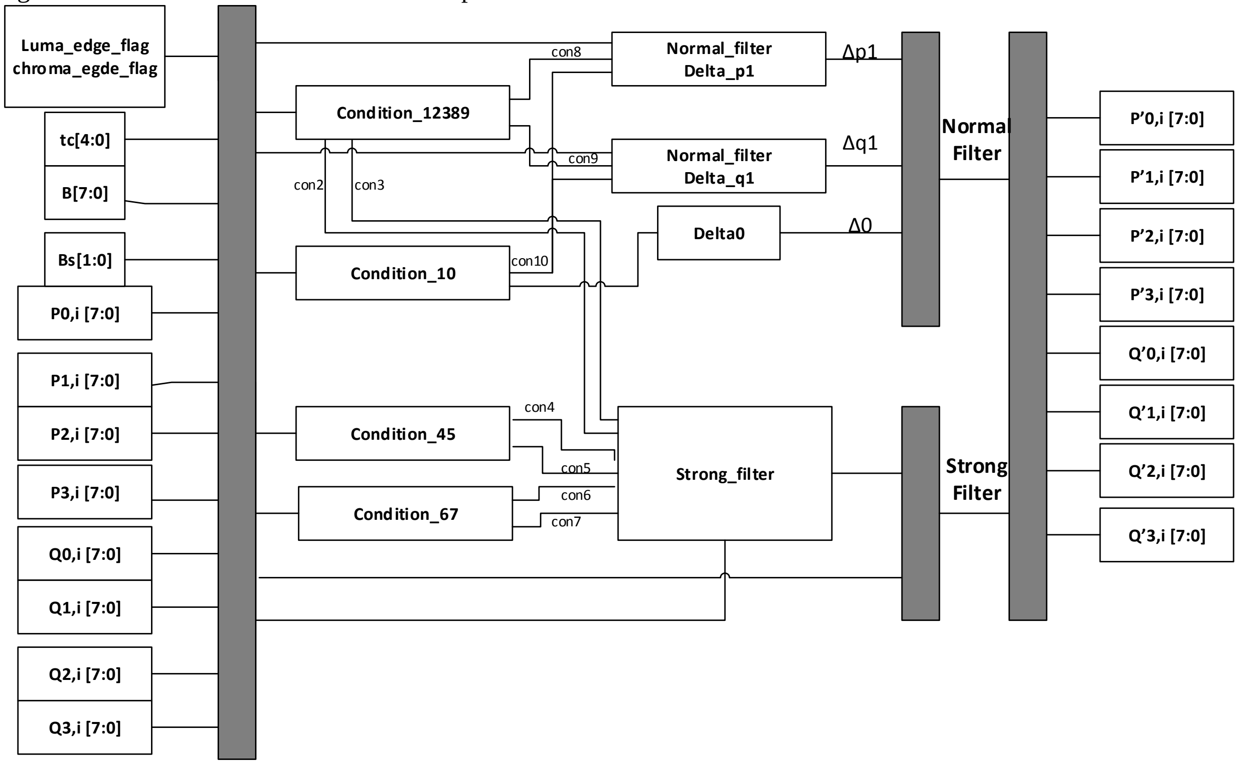

Fig. 6 The block diagram of module filter

We assume that, filtering operation is applied for the edge between CU P and CU Q. Module filter will decide whether to filter or not and the filter mode operation (strong filter or weak filter). First of all, value of Bs must be checked, if Bs =0 no filtering operation is performed. When Bs > 0, the following condition is checked to decide whether the deblocking filter is applied or not:

$$| p_{2,0} – 2*p_{1,0} + p_{0,0} |+| p_{2,3} – 2*p_{1,3} + p_{0,3} |+| q_{2,0} – 2*q_{1,0} + q_{0,0} |+ | q_{2,3} – 2*q_{1,3} + q_{0,3} | < \beta \quad (1)$$

where the threshold β depends on QP. Only the first and the fourth lines in a block boundary of length 4 are evaluated to avoid unnecessary complexity. If condition (1) is true, the filter is applied to the edge, otherwise not. Whether to apply a strong or weak De-blocking is also determined based on the first and the fourth lines across the block boundary of four samples. The following conditions are investigated to determine the filter mode operation:

$$| p_{2,0} – 2*p_{1,0} + p_{0,0} |+| q_{2,0} – 2*q_{1,0} + q_{0,0} | < \beta/8 \quad (2)$$
$$| p_{2,3} – 2*p_{1,3} + p_{0,3} |+| q_{2,3} – 2*q_{1,3} + q_{0,3} | < \beta/8 \quad (3)$$
$$| p_{3,0} – p_{0,0} | + | q_{0,0} – q_{3,3} | < \beta/8 \quad (4)$$
$$| p_{3,3} – p_{0,3} | + | q_{0,3} – q_{3,3} | < \beta/8 \quad (5)$$
$$| p_{0,0} – q_{0,0} | < 5tc/2 \quad (6)$$
$$| p_{0,3} – q_{0,3} | < 5tc/2 \quad (7)$$

If these conditions are true, the strong filter is selected. Then p_0 to p_2 and q_0 to q_2 are the filtered pixels, and the filter operations are as follows:

$$p_0' = (p_2 + 2*p_1 + 2*p_0 + 2*q_0 + q_1 + 4) >> 3$$
$$q_0' = (p_1 + 2*p_0 + 2*q_0 + 2*q_1 + q_2 + 4) >> 3$$
$$p_1' = (p_2 + p_1 + p_0 + q_0 + 2) >> 2$$

$q_1' = (p_0 + q_0 + q_1 + q_2 + 2) >> 2$
$p_2' = (2*p_3 + 3*p_2 + p_1 + p_0 + q_0 + 4) >> 3$
$q_2' = (p_0 + q_0 + q_1 + 3*q_2 + 2*q_3 + 4) >> 3$

Otherwise the weak filter is selected. Then there are three conditions is investigated:

$| p_{2,0} – 2*p_{1,0} + p_{0,0} |+| p_{2,3} – 2*p_{1,3} + p_{0,3} | < 3β/16$ (8)
$| q_{2,0} – 2*q_{1,0} + q_{0,0} |+| q_{2,3} – 2*q_{1,3} + q_{0,3} | < 3β/16$ (9)
$| Δ_{0,I} | <10tc\ (0<= i<=3)$ (10)

If condition (10) is true the modified value p_0' and q_0' are calculated as follows:

$p_0' = p_0 +Δ_0$
$q_0' = q_0 –Δ_0$

If condition (8) holds the modified p_1' is calculated as follows:

$Δp = Clip3(-(t_C >> 1), t_C >> 1, (((p_2 + p_0 + 1) >> 1) – p_1 + Δ) >>1)$
$p_1' = Clip1_Y(p_1 + Δp)$

Likewise, if condition (9) holds, q_1' is calculated as:

$Δq = Clip3(-(t_C >> 1), t_C >> 1, (((q_2 + q_0 + 1) >> 1) – q_1 – Δ) >>1)$
$q_1' = Clip1_Y(q_1 + Δq)$ (11)

IMPLEMENT RESULT

A hardware architecture of parallel De-blocking filter for 32×32 luminance CTU has been presented. The design has been synthesis in Xilinx Virtex 7 FPGA chip XC7VX330T-3FFG1157, using 18109 LUT, 4142 register and 178 blocks RAM/FIFO, and maximum frequency up to 113 Mhz. Therefore, throughput of this design is 38 Gbps. Table 2 presents a summary of the throughput of this design, it also presents results from [4], [5], [6] and [7] for comparison. You can see that, in our design, the maximum throughput is formidable than other design, of course hardware usage is also much higher than others but in HEVC standard, this not a problem. By a simple calculation, we can prove that our design can support for video application up to 4Kx2K (4096 x 2048), 30fps at frequency approximate 100MHz.

Table - 2 Implementation Comparisons

Design	*Throughput (Gbps)*
Shen [4]	14,9
Dalcin [5]	12,8
Huong [6]	20
Ozcan [7]	22,6
Proposed design	38

CONCLUSION

In this paper, we presented a high throughput and parallel architecture of De-blocking Filter for HEVC standard. The proposed architecture is based on a new parallel processing order which presented in [6]. The throughput of this design up to 38 Gbps, the proposed architecture can be suitable with high performance and high resolution video application, i.e HD or Ultra HD. Result obtained in this paper is the beginning, when the project completed, it will certainly achieve remarkable results.

REFERENCES

[1] B Bross, WJ Han, GJ Sullivan, JR Ohm and T Weigand, High Efficiency Video Coding (HEVC) Text Specification Draft 9, *ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) document JCTVC-K1003*, **2012.**

[2] Telecommunication Standardization Sector of ITU, High Efficiency Video Coding (HEVC), **2013.**

[3] Andrey Norkin, Kenneth Andersson, Arild Fuldseth and Gisle Bjontegaard, HEVC De-Blocking Filtering and Decisions, *Applications of Digital Image Processing XXXV. Proceedings of the SPIE*, **10/2012**, 8499, Article ID, 849912.

[4] Weiwei Shen, Qing Shang, Sha Shen, Yibo Fan and Xiaoyang Zeng, A High-Throughput VLSI Architecture for De-blocking Filter in HEVC, *IEEE International Symposium on Circuits and Systems*, Beijing, **2013,** 673 – 676.

[5] Felipe Vogel Dalcin, A De-Blocking Filter Architecture for High Efficiency Video Coding Standard, *Graduate Thesis*, Porto Alegre**, 2014.**

[6] Hoai Huong Nguyen Le and Jong Woo Bae, High-Throughput Parallel Architecture for H265/HEVC De-Blocking Filter, *Journal of Information Science and Engineering*, **2014,** 30, 281-294.

[7] Erdem Ozcan, Yusuf Adibelli and Ilker Hamzaoglu, A High Performance De-Blocking Filter Hardware for High Efficiency Video Coding, *Consumer Electronics, Transactions IEEE on*, **8/2013**, 59(3), 714-720.