International Journal of Control and Automation
Vol. 8, No. 6 (2015), pp. 275-282
http://dx.doi.org/10.14257/ijca.2015.8.6.27

A Method based on H.264 to de-block Filter Hardware System

Zhang Xiaohui[1]*, Yang Hongping[2], Shen Shanlan[3], Zang Haihe[1] and Zhang Zhixia[4]

[1]College of Electrical Engineering, Henan University of Technology, Zhengzhou 450001, China
[2]School of Engineering, Tianshui Normal University, Tianshui, 741001, China
[3]NanYang Technological University, 639798, Singapore
[4]School of Information Engineer, Henan Institute of Science and Technology, Xinxiang 453003, China
**Correspondence author: xiaohui.xh@163.com*

Abstract

H. 264 / AVC is a kind of newest video standards, the use of all kinds of complex forecasting mechanism, makes the compression ratio greatly to ascend, which provides a reliable algorithm support for high-definition video real-time transmission. In this paper, in order to balance boundary treatment of data dependency relationship and macro block filtering the speed of contradictions, a kind of H.264 to block filter hardware structure in SMIC0 was proposed. Under the library 0.13μm CMOS, the comprehensive results show that the circuit in the 300 MHZ clock frequency consumption 40000 logic gates, which needs 64 clocks cycle to deal with a 16 x16 macro block.

Keywords: *H.264, De-blocking, Hardware Architecture*

1. Introduction

Standard h. 264 / AVC video algorithm, can provide high-definition video and monitor application fields such as high compression ratio, high adaptability, high quality quality of solutions. Due to various reasons, however, reconstructing the frame can produce block effect, to block filter system can effectively solve this problem.

Deblocking filtering [1] is an important characteristic of H. 264/AVC video coding standard [2]. It not only can eliminate the blocking effect and improve the visual quality, but also reduce the bit rate of 5~10% [3-4]. It is critical for effective realization of high-resolution video applications. Although deblocking filtering can improve the coding efficiency, yet it requires higher storage costs and compute complexity. These requirements limit the deblocking filtering to real-time achievement, especially for high-resolution video applications. The literatures [5-8] proposed a lot of effective deblocking filtering hardware. Some of the structures use a single filtering unit and the horizontal boundaries of the vertical boundary processing [5]; some parallel filtering structures use vertical and horizontal structure boundary simultaneously, to improve throughput at the same time to increases the area [6]. Aiming at high-definition video 1920x1080 (1080p) for real-time coding, considering the two factors of speed and area. In the proposed structure of the filter circuit, the 48 boundary were water processed in order to proposed method. Balancing the dependence of the boundary of the data processing of the macroblock (MB) filtering conflict overall speed of such filtering is only 64 MB of clock cycles. In addition, the filter calculation pixel sample line can be divided into three parts, rational planning of the data flow, the process can be a good balance between the various stages of the operation.

ISSN: 2005-4297 IJCA

Comprehensive experimental results under SMIC 0.13μm CMOS technology library show that this circuit is clocked at 300MHz road consume 40,000 logic gates, and the processing speed up to 36frame/second, which fully meet the needs of real-time encoding of HD video.

2. Deblocking Filtering Algorithm

2.1. Algorithm

Since the conversion and motion estimation are the smallest size 4x4. Therefore, except for the image boundary, all the 4x4 block boundary luminance and chrominance blocks are required spent filter. The deblocking filtering process includes filtering horizontal vertical boundary and the horizontal boundary of the vertical filter. According to the standard of H.264/AVC, filtering based on MB conducted in accordance with the raster scan order of the MB image processing sequence. For each brightness MB, firstly, the leftmost vertical boundary MB was filtered. Then, the other three internal vertical boundaries were filtered from left to right. Similarly, the top boundary of the first MB was filtered, and then three internal horizontal boundaries were filtered from top to bottom. For each boundary of the 8x8 chroma MB, we use the similar sequence to chroma filter.

Table 1. The Boundary Strength BS

Boundary strength	The conditions what blocks p and q meet
4	One is intra-coded and the boundary belongs to macro block boundaries
3	One is intra-coded and the boundary doesn’t belong to macro block boundaries.
2	Neither is intra-coded; one contains a non-zero coding coefficient.
1	Neither is intra-coded; neither contains non-zero coding coefficient; different reference frame; the motion vector difference is less than 4.
0	The above conditions are not met.

Boundary strength (Boundary Strength, BS) controls the filter strength of 4x4 blocks layer, which range from 4 (strong filter) to 0 (no filtering). The values of BS are 1, 2 and 3, pixel block boundary filtering process using weak filtering. In the standard of H.264/AVC, the deblocking filter to the reconstructed luminance of each 16x16 MB as a filtering unit for each boundary, and the horizontal or vertical lines are processed pixel 16 (0-15). Luminance and chrominance corresponding position on the boundary of the BS share the same value. BS value determination process is shown in Table 1. The two boundaries of 4x4 neighboring blocks corresponding to the actual pixel sample line 4. Figure 1 shows a vertical or horizontal row of pixels between adjacent blocks p and q, each pixel row including eight pixels between 4x4 blocks (p3, p2, p1, p0, q0, q1, q2, q3). For vertical and horizontal block boundary filtering, which is actually four horizontal rows and vertical pixel data are processed and updated.

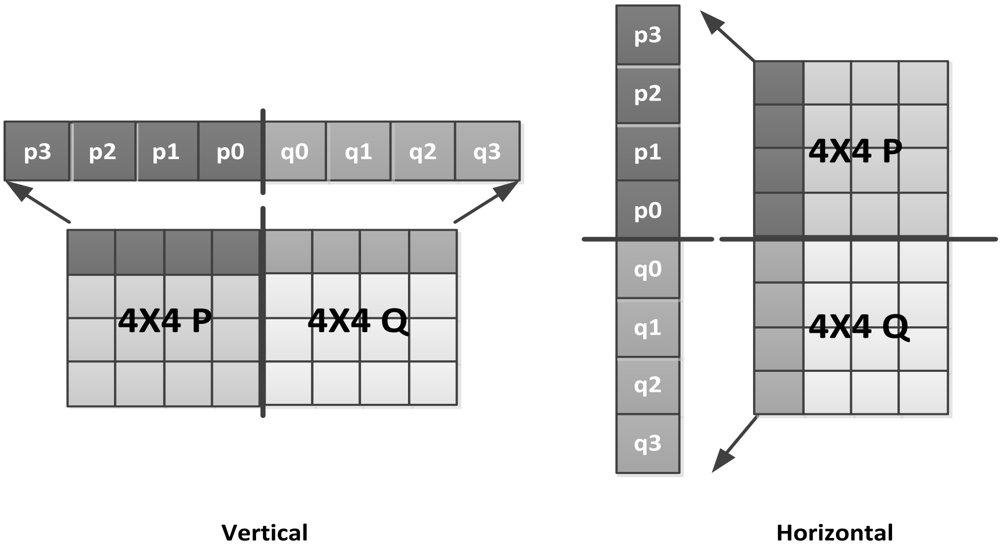

Figure 1. A Vertical or Horizontal Row Of Pixels between Adjacent Blocks P and Q

In fact, H. 264/AVC deblocking filtering adaptive low pass filter to update the contents of the pixel rows. For each boundary, according to the BS, the threshold values α and β (H.264 is a set of two false judgment threshold boundary), and the contents of the selected row of pixels suitable filter. Specific steps are as follows:

(1) Distinguishing the true and false boundary. When the following conditions are met, then the current boundary is not the true boundary.

$$|p0 - q0| < \alpha(\text{IndexA}) , \tag{1}$$

$$|p1 - q0| < \beta(\text{IndexB}) , \tag{2}$$

$$|q1 - q0| < \beta(\text{IndexB}) . \tag{3}$$

(2) Determining the filtering range. Namely, to decide pixels in what positions need to be processed and updated in the pixel sample line. The details are as follows:

①When BS = {1, 2, 3}, use a 4-tap linear filter. When the inputs are p1, p0, q0 and q1, the filter outputs are P0 and Q0. As for the luminance component; if | p2－ q0| ＜ β, using a 4-tap linear filter and giving the inputs of p2, p1, p0 and q0, the filter output P1 is produced. Similarly, if | q2－ q0| ＜ β, using a 4-tap linear filter and giving the inputs of q2, q1, q0 and p0, the filter output Q1 is produced.

②When BS = 4, if the formulae (4) and (5) are workable, use the 4-tap filter to calculate p1 and q1, and use a strong 5-tap filter for p0, q0, p2, and q2. If the formulae (4) and (5) do not hold, it is only necessary to calculate p0 和 q0 with a 3-tap filter so as to achieve the purpose of correction.

$$|p2 - q0| < \beta \& |p0 - q0| < (\alpha >> 2) + 2 \tag{4}$$

$$|q2 - q0| < \beta \& |p0 - q0| < (\alpha >> 2) + 2 \tag{5}$$

③When the degree block is under such situations as BS varying from 1 to 4, it is only to modify the sample –point values of p0 and q0.

2.2. The Original Filter Order

In the algorithm of H.264 standard, the filter order is always follow the first-left-first-top principle. That means, the whole filter system should process the first top-left 4x4 block, then the second top-left 4x4 block, and so on. Each 4x4 block contains 4 margins, and each 16x16 macro block contains 64 margins. If we have to follow the standard, we should filter every edge we need, totally 64 edges.

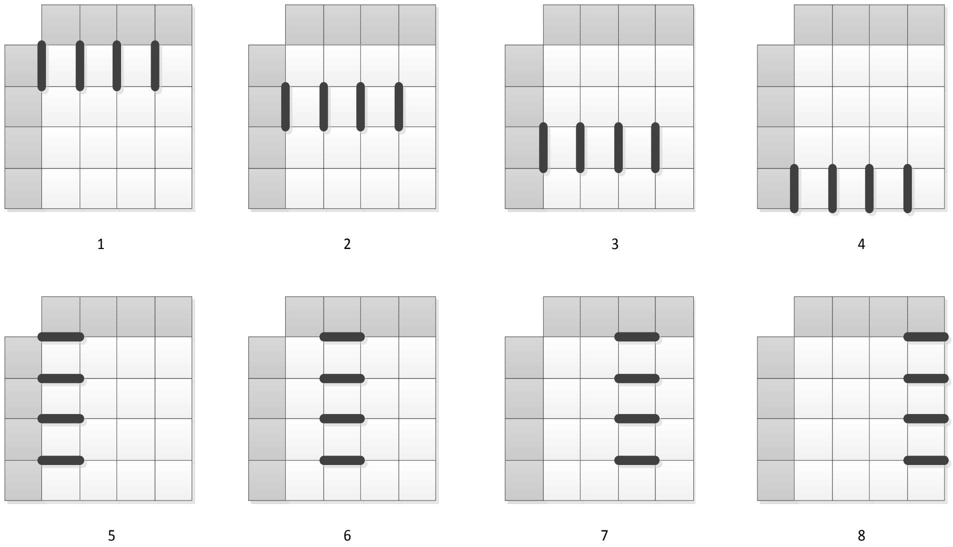

Figure 2. The Original Filter Order

The filter order is shown as Figure 2. First step, when we were doing process in first row, it only read out the first row 4x4 block from left to right, one by one. If one block was finished with filtering, it would be stored back to the buffer. Noteworthy, it only filtered one left edge. Then the second row 4x4 blocks are taken out from buffer, one by one either. It should be noticed that every block needs the neighbor 4x4 block information to be a reference for filtering the current block. For example, if we want to filter the first block left edge of the second row, we need read out the left neighbor block of the current 4x4 block. And if we want to filter the next 4x4 block, we also need read out the first block of the second row. All vertical edge always follow this principle, the horizontal edge is also same and a little difference for that is from top to bottom.

Now, we can comprehend the original filter order, and more clearly realize that filtered each edge always needs read out the ‘last’ block (vertical, left; horizontal, top). Each 4x4 block contains 4 edges, so that means each 4x4 block would be read out from buffer 4 times. If the total 16 4x4 blocks were finished with filtering, we need do the 64 operation times for buffer.

3. Hardware Design of Deblocking Filtering

Input data of the deblocking filter are mainly three kinds of judgment required filter strength parameter data, and the pixel value to be filtered control signals. The output is the pixel value after filtering. Complete control of the main control circuit of the entire work of the deblocking filter module, between the data transmission and the operation of each module [8-10]. The system structure is shown in Figure 3. Wherein, the filter calculation is the core of the whole system.

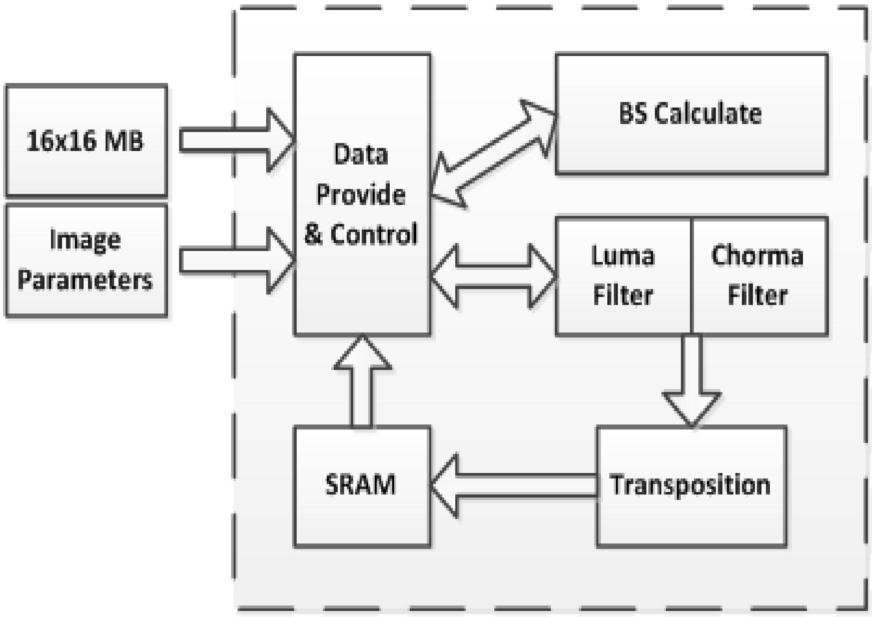

Figure 3. De-Block Hardware System

3.1. The Optimized Filtering Order

From the antecedent section, we noticed that the operation times for buffer we need is reach to 64. It is an uncanny operation frequency for memory. Most of performance are wasted by doing this to lead to reduce the whole system efficiency. If we want to design an efficient system, we must design an optimal filtering first. Especially, we should focus on the filter order. To improve the speed of the entire system, this paper proposes a new filter calculation sequence. This filter is calculated using a pipelined architecture. Filtering order is carried out in accordance with the order from as shown in Figure 4.

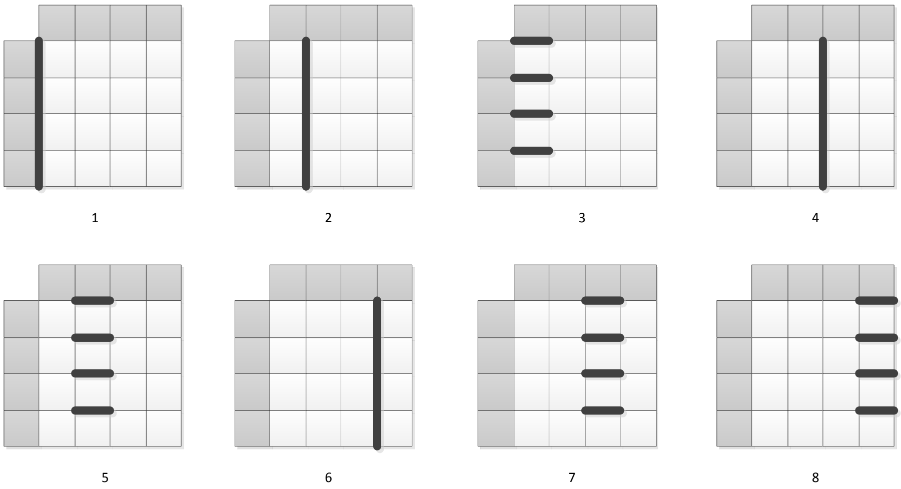

Figure 4. Filtering Order

The sequence as shown in Figure 4, first filtered the first vertical boundary, when the completion of the first vertical boundary filtering, the second vertical boundaries was filtered, at this time the first filtered boundary of four 4x4 blocks will be cached. When the second vertical boundary starts of a macro-block, then the level of the boundary of the first column enter into the filter simultaneously, and start filtering, and so the system can be no stagnant water level filtering. The traditional filtering mode, there need to wait until after the end of all levels of filtering, and then remove out of the finished macroblocks from storage unit again, and then to vertical filter, so that increase the number of visits to the storage unit, and reduce the efficiency of filtering. According to the calculation sequence of the conventional filter, filtering a 16X16 macroblock requires repeated read and write operations to the storage unit 64 times in the worst case, but the use of the proposed method only requires 32 times, reducing 50% of the number of operations. Chrominance blocks are also filtered in this way. Meanwhile, this method is more efficient for design ASIC pipeline because of reducing the memory operated times.

3.2. A Filter Processing Unit

Filtering calculation is the core of the whole filter module unit, the structure diagram of filtering calculation module unit as shown in Figure 5. QP value is calculated according to the look-up table out limits. At the same time, the need for filtering pixel value information (including information on the current block and adjacent blocks) are entered into the cache, the size of the input pixel data according to the filtering strength of the filter to be processed in the computing module, and then filtered according to the filtering strength threshold and select the appropriate filter calculation model, the final output pixel value after the re-deposited in the filter calculation in the SRAM, for subsequent operation.

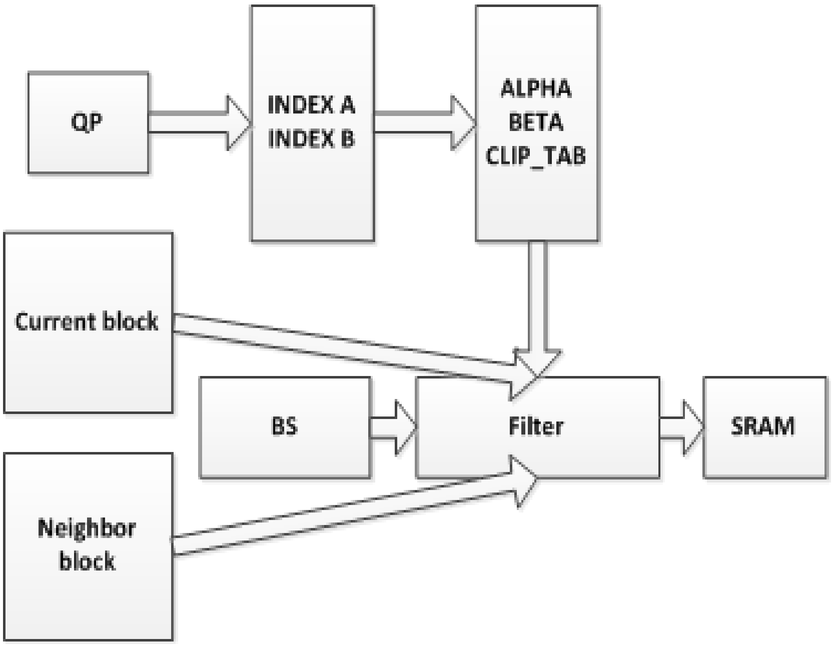

Figure 5. Filtering Computing Hardware Module

3.3. Transposition Module

We also noticed the difference between the horizontal filtered and vertical filtered. As the direction is discrepancy, we need an additional judgment part to control the filtering direction. For module reuse, in this paper, we designed the transposition module to fix that.

Figure 6 is schematic diagram of transposition. After all of 16 4x4 blocks first vertical filtered are finished, they will go through the transposition module to transpose. And then, they will be filtered the second time. It reuse the filtering module and reduce the design scale.

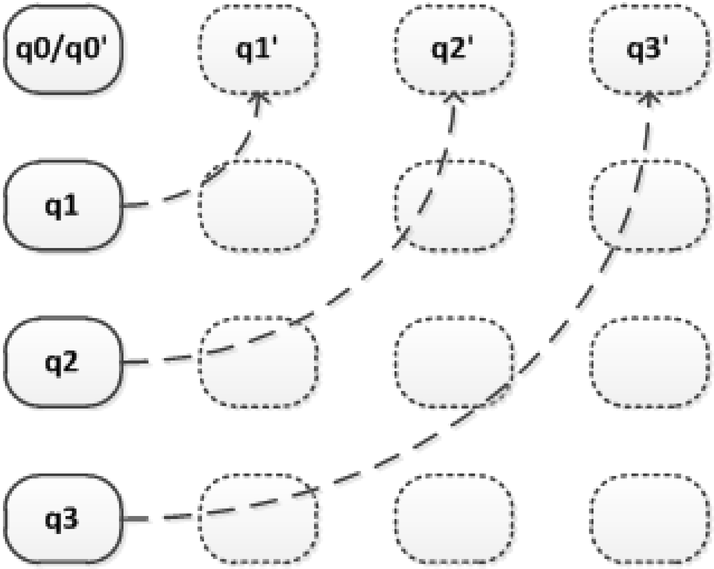

Figure 6.Transposition Schematic Diagram

4. Experimental Results and Analysis

720p series 720pstockholm_ter, the local filtering results before and after through this article filtering system are shown in Figure 7.

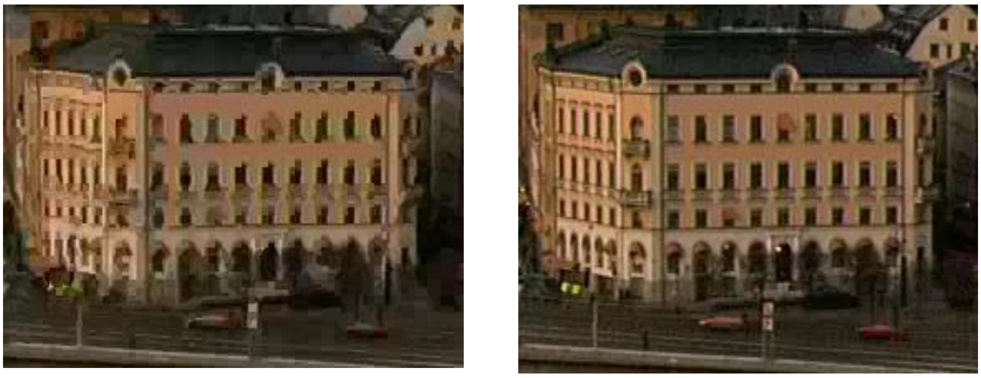

Figure 7. The Local Filtering Results Contrast Figure

Through Verilog HDL on the description of deblocking the proposed filtering structure and verification under VCS environment, compiled simulation, using SMIC process library 0.13μm, to synthesis by the compiler of Synopsys' design. When the operating frequency is set 300MHz, the maximum path delay of the circuit is 2.53ns, and the circuit consumes 40,000 logic gates. After testing, this paper designed the structure can work correctly, stability and support real-time applications. In this paper, the structure of a macro block filter requires a total of 64 clock cycles for the 4:2:0 sampling format, with a resolution of 1920x1080 ultra high-definition video, the processing speed can reach 36fps. Comparative analyses of the experimental results are shown in Table 2.

Table 2. Analysis of the Results

Article	Image size	Technology（μm）	Logic gates	Processing speed /cycles
[5]	1080P	0.18	20.8K	224
[6]	1080P	0.18	17.45K	504
[7]	1080P	0.18	20.3K	232
This paper	1080P	0.13	42K	64

It can be seen from Table 2, due to the structure of the proposed filter is four 4x4 blocks parallel filtering, and in [5]-[7], the number of gates higher about 1 times compared to this paper, but it was able to reach 64 clock cycles to process a 16x16 macroblock, speed can be increased 71%, while 50% reduction in the number of memory accesses.

5. Conclusion

By simulation and comprehensive, and the experimental data show that, this article design and optimization of VLSI structure, loop filter in SMIC0.13, the comprehensive results show that the circuit in the 300 MHZ clock frequency consumption 40000 logic gates, which needs 64 clocks cycle to deal with a 16 x16 macro block.so it can fully meet the 1080 p video real-time encoding (30 frames per second).

Acknowledgments

This study was funded by the Science and Technology Research Key Project of Henan Education Department (14B416001) and Henan University of technology high-level talent funded project (2013BS063).

References

[1] P. List, A. Joch and J. Lainema, “Adaptive Deblocking filter”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7, **(2003)**.

[2] Video TEAM Joint, ISO/IEC 14496-10 AVC Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification, S.Geneva: Telecommunication Standardization Sector of ITU, **(2005)**.

[3] T. Wiegand, G. J. Sullivan and G. Bjontegaard, “Overview of the H. 264/AVC video coding standard”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7, **(2003)**.

[4] L. Yunhong, Z. Long and L. Jihong, “Research on the infrared image enhancement method”, Journal of Xi’ an Polytechnic University, vol. 23, no. 4, **(2010)**.

[5] C. M. Chen and C. H. Chen, “Configurable VLSI architecture for Deblocking filter in H.264/AVC”, IEEE Trans. Very Large Scale Integration System, vol. 16, no. 8, **(2008)**.

[6] Y.-W. Huang, T.-W. Chen, B.-Y. Hsieh, T.-C. Wang, T.-H. Chang and L.-G. Chen, “Architecture Design for De-blocking Filter in H.264/JVT/AVC”, International Conference on Multimedia and Expo(ICME’03), **(2003)** July.

[7] M. Sima, Y. Zhou and W. Zhang, “An Efficient Architecture for Adaptive Deblocking Filter of H.264/AVC Video Coding”, IEEE Transactions on Consumer Electronics, vol. 50, no. 1, **(2004)** February.

[8] T. liu and W. Lee, “An In/Post-loop Deblocking Filter with Hybrid Filtering Schedule”, IEEE Transaction on Circuits and Systems for Video Technology, **(2007)** July.

[9] S.-C.Chang, W.-H. Peng, S.-H. Wang, and T. Chiang, “A platform based bus-interleaved architecture for de-blocking filter in H.264/MPEG-4 AVC”, IEEE Trans. Consum. Electron, vol. 51, no. 1, **(2005)** February.

[10] P. P. Dang, “An Efficient Implementation of In-loop Deblocking Filters for H.264 Using VLIW Architecture and predication”, IEEE Transaction on Consumer Electronics, **(2005)**.