DOI: 10.21917/ijme.2015.0002

AN EFFICIENT ARCHITECTURE FOR DE-BLOCKING FILTER

Meghavi H. Modi[1] and Nehal N. Shah[2]
Department of Electronics and Communication Engineering, Sarvajanik College of Engineering & Technology, India
E-mail: [1]meghavimodi123@gmail.com, [2]nehalsh@gmail.com

Abstract

H.264 standard uses block based motion estimation, motion compensation, transform and quantization processes to perform video compression. By the use of block based operations, it would result into the discontinuity at block boundary-known as blocking artifacts. In this paper, high throughput filter architecture for real-time implementation of de-blocking filter is presented which reduces memory access requirements and results in less clock cycles, to process a macroblock. Post processing approach is used in the paper in order to reduce the blocking artifacts and hence for reducing complexity of the architecture. The proposed architecture design uses only 23 clock cycles to process a single macroblock and in addition, the architecture effectively utilizes the buffers to store the intermediate data. The operational frequency of the proposed architecture is 55.675 MHz. The proposed architecture is implemented in VHDL and synthesize for Xilinx FPGA. It can process 75 HD frames with 1920 × 1080 resolutions.

Keywords:

De-blocking Filter, H.264/AVC, Motion Estimation, FPGA, Macroblock, VLSI Architecture Design

1. INTRODUCTION

Video compression is important topic in today's research in the field of multimedia like digital television broadcasting, Internet video streaming, Mobile video streaming, and Video calling. Without compression it is difficult to transmit videos over a network or to store due to the large amount of information that they contain. As shown in Fig.1, all current video compression standards including MPEG-1/2/4, H.261/2/3/4, performs block based motion estimation (ME), motion compensation (MC), transform and quantization processes to perform compression. The efficient compression is achieved by motion estimation and block based integer transform. The use of block-based processing introduces artifacts on the block edges. The de-blocking filter (DBF) is used to reduce the blocking artifacts, which increases coding efficiency and improves the decoded video quality.

As shown in Fig.1, the de-blocking filter is used after the inverse transform and quantization process to improve the objective and subjective quality of decoded frames. Two approaches are often used in order to perform filtering for reducing blocking artifacts i.e. in-loop filter [2], [3], [4] approach and post filter [5], [6] approach. In-loop filter is a part of encoder while post filter only operates on the display buffer, outside of the coding loop. In-loop filter forces all standard decoders to perform identical filtering in order to stay in synchronization with the encoder and increases the computational complexity but the post-processing requires no modifications of existing standards to get better quality hence it is the most practical solution to remove the blocking artifacts. In this paper, the filter unit with post filter approach is used.

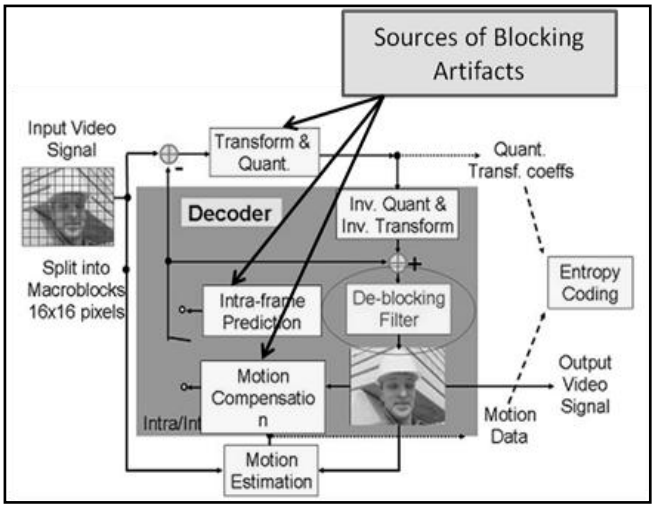

Fig.1. H.264 encoder with sources of blocking artifacts [1]

The different de-blocking filter architectures have been implemented based on various parameters like processing order of the macroblocks, memory organization and utilization using single port [7], [8] or dual port [2], [4] memory, and in-loop filter approach. Double filtering strategy with serial processing order has been defined in [2] where horizontal as well as vertical filtering has been performed simultaneously on a macroblock. High throughput hybrid filtering schedule with in-loop/post approach has been discussed in [3] where intermediate pixels are reused when de-blocking filter switches the filtered edges from vertical to horizontal direction. Architecture with double cross filtering process and with in-loop approach where horizontal filtering in two directions and vertical filtering in two directions i.e. four parallel filtering are performed simultaneously is discussed in [4]. Architecture presented in [2], [3], [4] computes boundary strength and demonstrate implementations. Algorithm presented in [5] and [6] is based on region classification among smooth, intermediate or complex and accordingly appropriate filter is applied.

Although the de-blocking filter can provide the improvement on coding efficiency, it requires high memory access and computational complexity. Due to such requirements, real time implementation of de-blocking filter is challenging especially for the high resolution video specifications. In this paper, classification and filtering mechanism of [5] is used and an efficient architecture of post de-blocking filter is implemented which works in real time. Section 2 describes the 8 × 8 macroblock boundary, activity based macroblock classification and filter units for smooth, complex, and intermediate macroblocks. Section 3 describes simulation and synthesis results and in section 4 conclusions is presented.

2. HARDWARE ARCHITECTURE FOR EFFICIENT FILTER UNIT

In the H.264 video coding standard, variable block size based operations are used in order to perform compression. Maximum block size is 16 × 16 which can be divided into 16 × 8, 8 × 16 or 8 × 8 size macroblock. 8 × 8 size macroblock can be divided into sub macroblock of 8 × 4, 4 × 8 or 4 × 4 block. This would result in discontinuity at not only at the block boundaries but also inside block. Due to that filter will be applied on all rows and columns on desired pixels. In order to effectively apply filtering, the whole video frame is divided into 8 × 8 size macroblocks as shown in Fig.2 and de-blocking block of size 8 × 8 is taken from the four adjacent macroblocks.

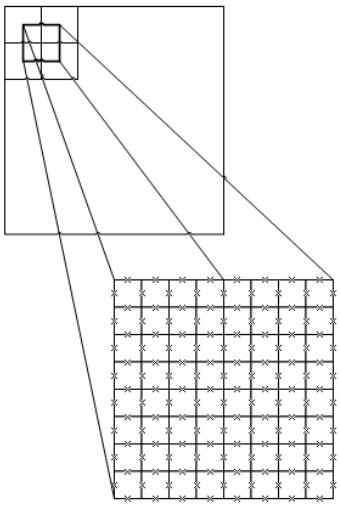

Fig.2. Macroblock selection from adjacent four blocks from the video frame

2.1 THE PROPOSED ARCHITECTURE

The proposed architecture consists of processing order unit, activity module, threshold table unit, horizontal filter unit (H-filter), vertical filter unit (V-filter), temporal buffer unit and transpose-inverse transpose unit. The basic block diagram of architecture is shown in Fig.3. Each macroblock is read from single port SRAM and stored in the temporal buffer of size 8 × 8 using input bus of 64 bits (8 pixels) wide. In order to efficiently reduce blocking artifacts, filtering is done on both directions. First horizontal filtering is done by H-filter unit and then data is transposed from row to column. The transposed data is given to V-filter unit for vertical filtering. During H and V filtering, all the threshold parameters are provided by the threshold table unit. Once the block is filtered in both the directions, it is sent back to SRAM through output bus and whole process is repeated for all the blocks.

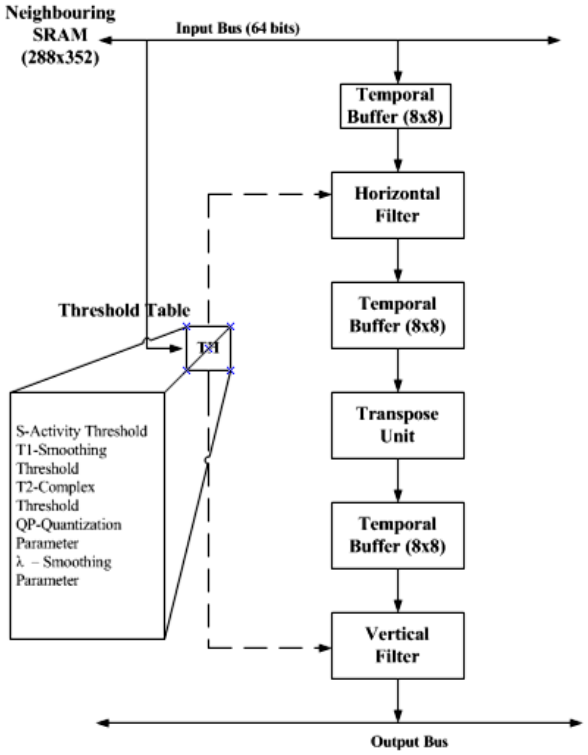

Fig.3. Block Diagram of Proposed architecture of De-blocking filter

2.2 ACTIVITY MODULE

In the video frames, the characteristics of various regions are different. Some of the regions of video frame consist of flat areas while some consist of textured areas. In such a case, same filtering is not appropriate for all the areas. So the filter unit is divided into three different regions with appropriate thresholds. Region classification is done with the help of activity module. The operation of activity module is shown in Fig.4 and its architecture is described in Fig.5. To begin with the absolute difference between the adjacent pixels is found out and each absolute difference is compared with the activity threshold S. If the difference is greater than threshold S, then output is 1 else the output is 0. Finally activity is calculated by the addition of each comparator's output and compared with smooth threshold T1 and complex threshold T2. If activity is less than T1, then smooth filter unit is applied on row else if activity is greater than T2, then complex filter unit is applied. If both these conditions are not satisfied then intermediate filter unit is applied. The same process is carried out for each row of 8 × 8 macroblock. To compute activity Eq.(1) and Eq.(2) are used

$$A(v) = \sum_{i=1}^{5} \Phi(v_i - v_{i+1}) \quad (1)$$

where,

$$\Phi(\Delta) = 0, \text{ if } |\Delta| \leq S \text{ else } 1 \quad (2)$$

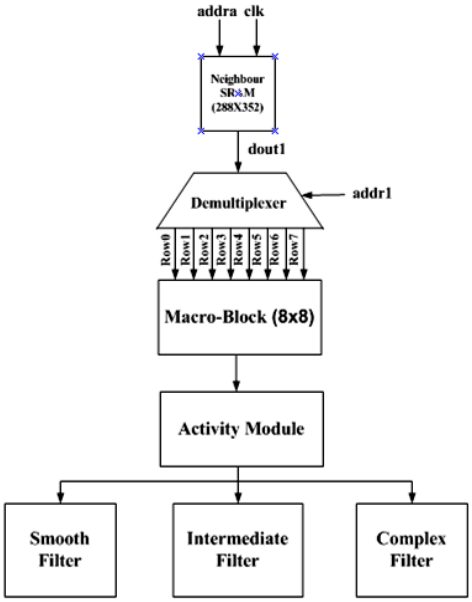

Fig.4. Detail flow architecture of Filter unit

2.2.1 Architecture of Smooth Filter:

For macroblock falling in smooth region, middle six pixels are modified as shown in Fig.6. The human visual system is more sensitive to blocking artifacts in flat region; hence more number of pixels is filtered. Offset-based filtering method is used in smooth regions and its implementation is shown in Fig.7. The offset is determined from the difference between two pixels *c* and *d* and the value of updated pixels are adjusted within the gray scale range 0 to 255. The difference is compared with quantization parameter (2 × QP). If the difference is less than (2 × QP), then only smooth filter is applied on macroblock. The pixels *a*, *b*, *c*, *d*, *e*, *f* are modified using offset as shown in Fig.8.

2.2.2 Architecture of Complex Filter:

If the blocking effects occur in a high activity region then four pixels *b*, *c*, *d* and *e* of Fig.6 are updated as shown in Fig.9. In this case if difference of pixels *c* and *d* is less than (QP) then only complex filter is applied on macroblock.

2.2.3 Architecture of Intermediate Filter:

For the intermediate filter unit, 3 × 3 window filter is considered as shown in Fig.10. It contains total nine pixels P1 to P9, where P5 defines the centre pixel of 3 × 3 window. Architecture for intermediate filter is shown in Fig.11. Absolute difference between the center pixel and adjacent pixels of window is found out and each difference is compared with quantization parameter (QP). Each output of comparator provides the alpha value ($\alpha 1$ to $\alpha 9$). S1 can be found out by addition of product of corresponding pixel value and alpha value. S2 can be found out by addition of output alpha. Modified centre pixel P5' is calculated with the help of S1, S2, P5 and smoothing extent parameter value λ as indicated in Eq.(4), Eq.(5) and Eq.(6) respectively.

$$\alpha_i = 1, \text{ when } |p5\text{-}p_i| < \text{Th otherwise } 0 \qquad (3)$$

Here Th is set according to current quantization parameter QP for intra coded macroblock and QP/2 for inter coded macroblock. Parameter λ controls the extent of smoothing and typically lies between 8 to16.

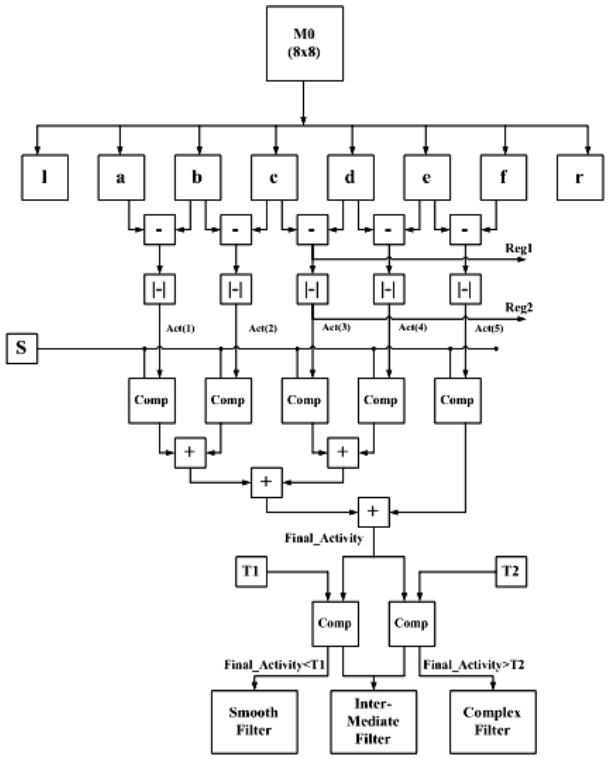

Fig.5. Architecture of activity Module

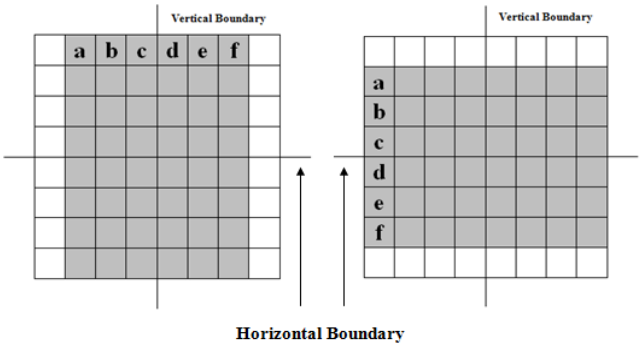

Fig.6. Filtering on smooth region [6]

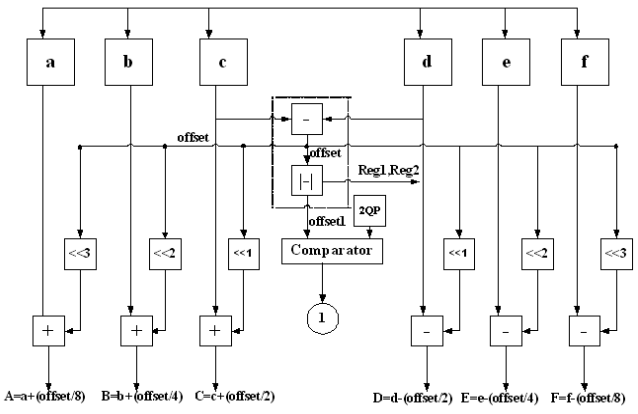

Fig.7. Architecture of smooth filter

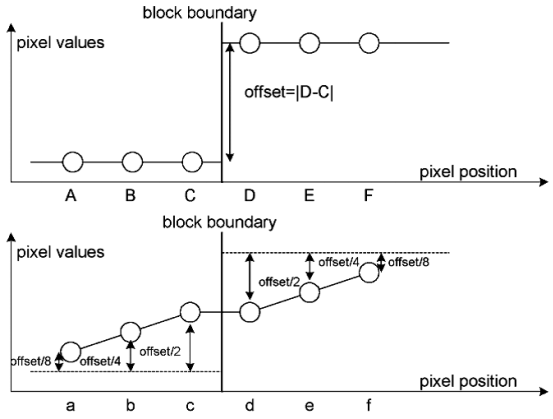

Fig.8. Filtering in smooth region on one row [6]

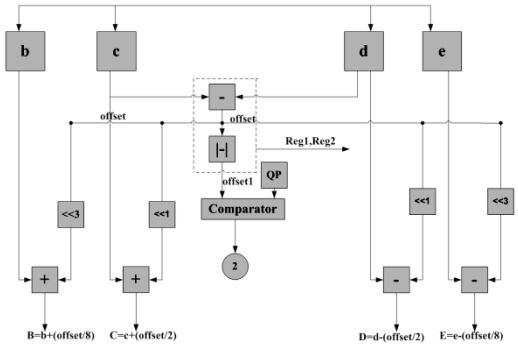

Fig.9. Architecture for Complex region

P1	**P2**	**P3**
P4	**P5**	**P6**
P7	**P8**	**P9**

Fig.10. Pixels in 3 × 3 for intermediate filter [6]

The same process is carried out for each row of macroblock. In this intermediate mode, a possible real edge must be preserved when the blocking effect is filtered.

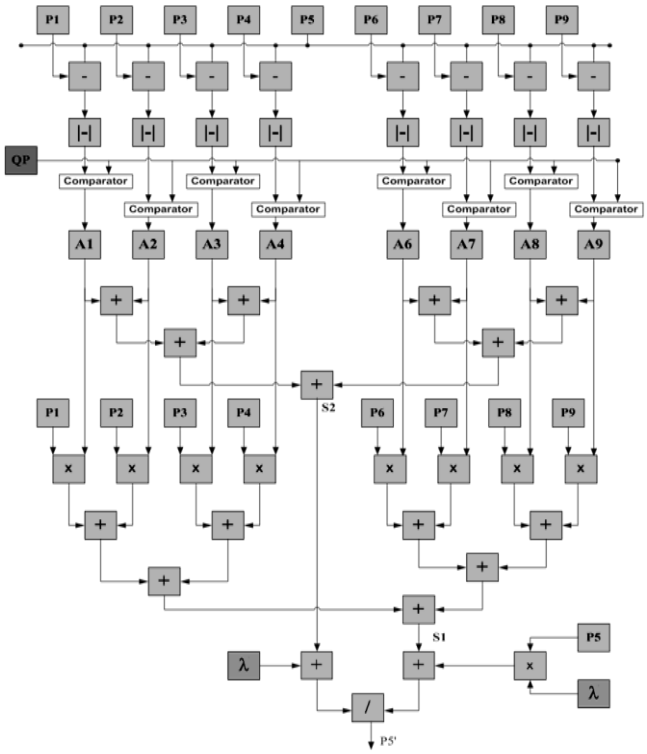

Fig.11. Architecture of Intermediate Filter

$$S_2 = \sum_{i=1, i\neq 5}^{9} \alpha_i \tag{4}$$

$$S_1 = \sum_{i=1, i\neq 5}^{9} \alpha_i P_i \tag{5}$$

$$P_5^{'} = \frac{(\lambda * P_5) + S_1}{\lambda + S_2} \tag{6}$$

2.3 CLOCK ANALYSIS

The Fig.11 describes the clock analysis of the proposed architecture. From the figure it is shown that, the first and second clocks are used as the latency. In the 2nd, 3rd and 4th clock cycles, first three rows of the macroblock are read. In the 5th clock cycle, the activity and offset of the respective row is calculated. During 6th to 13th clock cycles, total eight row of a single macroblock are horizontally filtered. As soon as the first row is filtered (at 6th clock cycle), the transpose of that row is carried out in next clock cycle (at 7th clock cycle). Once all rows are transposed, in next 8 clock cycles, vertical filtering is applied on them. In the same way, all the blocks are filtered and stored back in SRAM.

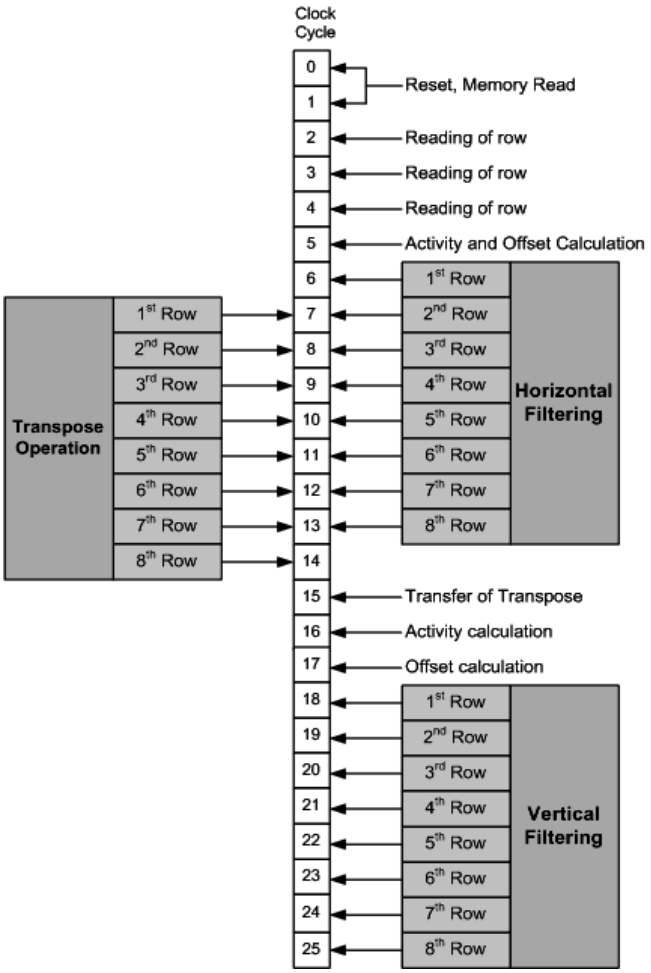

Fig.12. Clock Analysis of Proposed Architecture

3. IMPLEMENTATION SETUP AND SYNTHESIS RESULTS

The implementation of the architecture is carried out using Xilinx 12.3 ISE software tool. CIF video frame (here Foreman frame) with blocking artifact is considered and stored in SRAM using IP core generator. De-blocking block is read from SRAM memory row wise. The activity threshold parameter S is taken as 2 value i.e. small enough in order to differentiate between flat (smooth) and complex area. Value of T1 and T2 are chosen as 2 and 3 respectively, λ is taken as 8 and QP is taken as 38. Activity is computed and first horizontal filter is applied. Block is transpose for vertical filtering which will beneficial as same filter architecture will work for both horizontal and vertical filter. Filtered macroblock is stored back in SRAM. After completion of all macroblock of given frame, PSNR is computed.

Architecture is synthesized for Xilinx vertex-5xc5vlx50-3ff676 FPGA. Most composite part of architecture is filter unit. Smooth, intermediate and complex filter unit consumes 68, 513, 43 LUTs respectively for filtering of one row of macroblock as shown in Table.1. Intermediate filter unit needs surround pixels from three rows hence need much more resources.

Table.1. Resources used by all three types of filter units

		Intermediate filter	Smooth filter	Complex filter
Logic Utilization	**Available**	**Used**		
Number of Slice Registers	28800	99	0	0
Number of Slice LUTs	28800	513	68	43
Number of fully used LUT-FF pairs	548	64	0	0

Table.2. Resource utilization of proposed architecture

Parameters	Used	Available	Utilization
Number of Slice Registers	1471	28800	5%
Number of slice LUTs	1757	28800	6%
Number of Block RAM /FIFO	22	48	45%
Frequency	55.675MHz		
PSNR	36.5845 dB		

As indicated in Table.2 proposed architecture uses 1757 LUTs and 1471 registers and work at 55.675MHz frequency. Architecture is compared with existing implementations in Table.3. Due to block based ME and MC process, blocking artifacts is spread inside block hence proposed architecture process 8 × 8 size macroblock which is middle portion of original 16 × 16 macroblock and provide better results for video quality. Compared to all other architecture it utilizes very few resources and can process 75 HDTV frames per second hence can be used for real time implementation. Results of foreman frame with and without blocking artifacts are indicated in Fig.13. Subjective frame quality is measured with PSNR which is 36.58dB.

4. CONCLUSION

Post de-blocking filter architecture is implemented in this paper which requires no modifications of existing standards and no synchronization is needed between the encoder and decoder. In order to reduce the complexity of architecture serial processing order is used which reduces the memory access requirements. The proposed architecture requires only 25 clock cycles in order to process the first macroblock and then with 23 clock cycles, the next macroblock is processed. Clock cycles required to process a single macroblock is less as compared to available single port architecture. Architecture outperforms other architecture in terms of resource utilization and throughput. It process 75 HDTV frames per second hence well suited for real time implementation.

(a). Original frame (b). Frame with blocking effect (c). Frame after filtering

Fig.13. Foreman frame with and without de-blocking filter

Table.3. Comparison of proposed architecture with existing implementations

Architecture	MB Size	Frequency	SRAM designs for pixels	Cycle/MB	Processing Order	Macroblocks processed /sec (10^5 MB/sec)	Gate Count (K)	FPS
Felix [2]	16 × 16	36.45MHz	Dual port 64 × 32	110	Sequential processing in horizontal direction	3.3	12.6	30 HDTV
Chen-Yi Lee [3]	16 × 16	100 MHz	Single port 96 × 32 × 2	243	Hybrid processing	4.11	21.1	30 CIF
TSung-Han [4]	16 × 16	46.6 MHz	Dual port 64 × 32	15	4-parallel processing	31.0	20.14	30 QFHD
Cheng [7]	16 × 16	100 MHz	Single port 80 × 31	336	Sequential Processing	2.9	9.16	30 QCIF
Huang [8]	16 × 16	100 MHz	single port 96 × 64 × 32	878	Sequential Processing in vertical direction	1.1	18.91	31.6 HDTV
Proposed	8 × 8	55.675 MHz	Single port SRAM 64 × 64	23	Sequential Processing in horizontal direction	22.2	3.2	75 HDTV 89 CIF

REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjontegaard and A. Luthra, "Overview of the H. 264/AVC video coding standard", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, No. 7, pp. 560–576, 2003.

[2] Félix Tobajas, Gustavo M. Callicó, Pedro A. Pérez, Valentín de Armas and Roberto Sarmiento, "An Efficient Double-Filter Hardware Architecture for H.264/AVC De-blocking Filtering", *IEEE Transactions on Consumer Electronics*, Vol. 54, No. 1, pp. 131-139, 2008.

[3] Tsu-Ming Liu, Wen-Ping Lee and Chen-Yi Lee, "An In/Post-Loop De-blocking Filter With Hybrid Filtering Schedule", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 17, No. 7, pp. 937-943, 2007.

[4] Tsung-Han Tsai and Yu-Nan Pan, "High Efficient H.264/AVC De-blocking Filter Architecture for Real-time QFHD", *IEEE Transactions on Consumer Electronics*, Vol. 55, No. 4, pp. 2248-2256, 2009.

[5] Shen-Chuan Tai, Yen-Yu Chen and Shin-Feng Sheu, "De-blocking Filter for Low Bit Rate MPEG-4 Video", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 15, No. 6, pp. 733-741, 2005.

[6] Jongho Kim, Minseok Choi and Jechang Jeong, "Reduction of Blocking Artifacts for HDTV using Offset-and-Shift Technique", *IEEE Transactions on Consumer Electronics*, Vol. 53, No. 4, pp. 1736-1743, 2007.

[7] Chao-Chung Cheng and Tian Sheuan Chang, "An hardware efficient de-blocking filter for H.264/AVC", *Proceedings on International Conference on Consumer Electronics*, pp. 235-236, 2005.

[8] 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", *Proceedings of International Conference on Multimedia and Expo*, Vol. 1, pp. 693-696 , 2003.