Original scientific paper

Informacije MIDEM
Journal of Microelectronics,
Electronic Components and Materials
Vol. 45, No. 1 (2015), 12 – 21

A Parallel Architecture with Novel Filtering and Data Accessing Order for Deblocking Filter in H.264/Svc Using Reconfigurable Architecture

I. Manju, A.Senthil Kumar

Velammal Engineering College, Tamilnadu, India.

Abstract: In this paper we present a parallel filtering architecture with novel filtering and data accessing order for deblocking filter in H.264/SVC. The deblocking filter is the complex part in H.264/SVC which consumes more computation time and it has to adapt for normal filtering (PAFF), MBAFF filtering and inter-layer prediction. The filtering order of MBAFF coded frames has to support all combinations of field/frame mode for current and adjacent MB to filter a macroblock which increases the complexity of deblocking filter. The proposed filtering architecture adapts efficiently for the MBAFF coded frames by reducing the complexity, results in faster filtering of a macroblock. Implementing the filter architecture in reconfigurable platform helps in faster adaptability to normal filtering operation and MBAFF filtering. The proposed deblocking filter architecture is implemented in Cyclone V (5CEFA9F31C8N) and the results are analyzed. The proposed architecture achieves 19% increase in processing speed and 21% reduction in area.

Keywords: H.264/SVC; Deblocking Filter; PAFF/MBAFF; Reconfigurable Architecture

Paralelna preoblikovalna struktura za deblokirni filter v H.264/Svc z novim filtriranjem in vrstnim redom dostopa do podatkov

Izvleček: V članku predstavljamo novo paralelno filtrno strukturo v H.264/Svc z novim filtriranjem in vrstnim redom dostopa do podatkov. Deblokirni filter je kompleksen del H.264/SVC, ki potrebuje več računskega časa in se mora prilagoditi navadnemu filtru (PAFF), MBAFF filtru in medslojnimi napovedmi. Vrstni red filtriranja MBAFF kodnih okvirjev mora podpirati vse kombinacije trenutnih in sosednih MB načinov polje/okvir za filtriranje makro bloka, kar zaplete deblokirni filter. Predlagana filtrna struktura se prilagodi MBAFF kodiranim okvirjem z zmanjšanjem obsežnosti, kar omogoča hitro filtriranje makro blokov. Predlagana struktura je implementirana v Cyclone V (5CEFA9F31C8N) in dosega 19 % višjo hitrost procesiranja in 21 % zmanjšanje površine.

Ključne besede: H.264/SVC; deblokirni filter; PAFF/MBAFF; preoblikovalna struktura

* *Corresponding Author's e-mail: drmanjujackin@gmail.com*

1 Introduction

H264/SVC is the recent international standard used for video coding [1]. It is a scalable video coding (SVC) extension of H.264/AVC standardized by the joint team of ITU-T VCEG and ISO/IEC MPEG. Due to these latest advancements in video coding standards it has been applied to various multimedia applications such as video telephony, video conferencing over mobile TV, Blu-ray Disc and HD DVD optical storage media [2-4], [15]. Nowadays RTP/IP is mostly used in modern video transmission and storage systems and it is characterized by variety of connection qualities and receiving devices [1]. The RTP/IP [16] access network is the standardized packet format for delivering audio and video over IP networks. The receiving devices are varied from cell phones to high-end PC's where variation is terms of both resolution and processing power of devices. H.264/SVC addresses these issues by providing scalable video sequence.

In H264/SVC [14], [10], [20] the scalability is in terms of spatial (resolution), temporal (frames) and quality (PSNR) by removing part of the video bit stream depending upon the need of the users. The scalability in

H.264/SVC is achieved by layered structures as base layer with several additional enhancement layers. The video performance is increased from base layer; with base layer is having lowest video content information. The deblocking filter employed in H264/SVC is of high complexity and consumes over 30% of total execution in H264/SVC. In H264/AVC [5], [6], [17] the in-loop deblocking filter is employed after motion compensation to remove the blocking artifacts. The block artifacts are resulted from both quantization of transform coefficient and block based nature of motion compensation. The H264/SVC employ the in-loop deblocking filter after the motion compensation for frames coded either in PAFF or MBAFF type and in the inter prediction layer of spatial resolution to remove blocking artifacts. In each case an adaptive deblocking filter [6], [11], [12] is applied on each 4x4 block edge considering the boundary strength (Bs) values of the pixel across the boundary based upon the block type whether it is intra or inter coded. The deblocking filter is implemented using various architectures [7-9], [18], [22]. In [9], a new filtering order which modifies the basic filtering order by adopting the data reusability between successive filtering. The filtering architecture in [8], achieves higher data reusability by combining both horizontal and vertical filtering of a 4x4 macroblock. Hybrid scheduling method in [7] uses less number of processing cycles to filter a macroblock. The same Hybrid scheduling method which uses both in/post-loop filters is effectively adopted for multiple standards H.264/MPEG 4 with reduced gate counts compared to other filtering architecture which supports multiple standards. In [21], scalable deblocking filter architecture provides parallelism at macroblock level in wave front order for filtering the frame. It is implemented in Virtex 5 and the level of parallelism is limited by the resource availability. In this paper, a novel filtering and data accessing order with parallel processing using reconfigurable architecture is adopted for the deblocking filter to support normal and MBAFF coded frames. By adopting a reconfigurable architecture using Cyclone V for these deblocking filter results in increase of computational speed and efficiency. Section 2 provides concept of deblocking filter. Section 3 gives clear explanation regarding the proposed deblocking filter architecture and its adaptability for PAFF and MBAFF coded frames with filter processing order. Sections 4 discuss the results obtained by implementing it in Cyclone V and compare it with various filtering architecture.

2 Deblocking Filter

In our architecture, an adaptive deblocking filter [2] is employed. The deblocking filter is used to remove the blocking artifacts resulted from both quantization process and motion compensation due to its block based nature. Each macroblock consist of one 16x16 luminance block and two 8x8 chrominance blocks. The deblocking filter is applied to each 4x4 block in the macroblock.

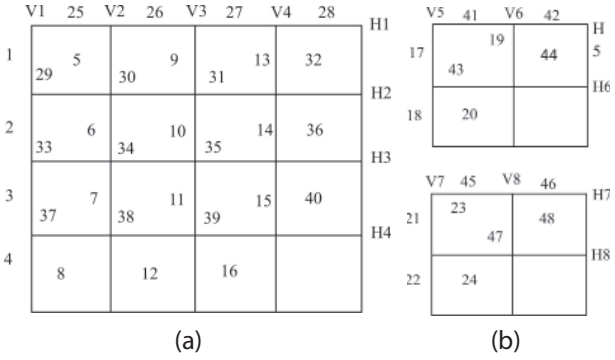


Figure 1: (a) luma block, (b) chroma block

The filtering is applied in the order of vertical edge first then on the horizontal edges as shown in Figure 1. The same filtering order is followed in chrominance block also. The deblocking filter is adaptive based on three levels they are slice level, edge level and sample level.

2. 1 Adaptability of Filter

2.1.1 Slice Level

In the Slice level, the $Offset_A$ and $Offset_B$ is transmitted along the slice header syntax which is used to adjust the values of α and β, which is a quantization dependent parameters. By varying the values of a and b from positive to negative, the filtering is varied from strong to weak compared to zero offset values. A zero offset value will give no change in filtering. A negative offset value will helps to maintain the edge sharpness in high resolution video.

Table 1: Bs value for each coded MB

Block nodes and conditions	Bs
One of the block is intra and its macro block edge	4
One of block is intra	3
One of the block has coded residuals	2
Different motion vector, Different Reference frame, Different no of reference frame	1
Otherwise	0

2.1.2 Edge level

The filtering applied for each 4x4 block depends upon boundary strength (Bs) value. The Bs value is varied from 4 to 0 based upon the block mode and the cod-

ing type of the two adjacent blocks with order of decreasing filter strength. The Bs value of 1 to 3 mentions standard filtering, value of 4 means strong filtering and value of 0 means no filtering. The varying filtering level reflects on the number of samples that has to be modified. In case of MBAFF, consideration has to be taken in applying a strong vertical filtering at the field level. The following Table 1 shows the boundary strength value for each coded block and filters that have been used.

2.1.3 Sample level

By using sample level adaptability in the deblocking filter, the original edges in the picture is preserved. The sample level adaptability is achieved by analyzing the values across the boundaries. Let P_0, P_1, P_2, P_3 and q_0, q_1, q_2, q_3 be the samples across boundaries of adjacent coded blocks. p_0 and q_0 be the sample at the boundaries. Figure2. shows the condition where filtering is applied. For Boundary strength (Bs) value other than zero, the following consideration has been taken in to account before applying filtering. The filtering for the line-of-pixels (LOP) will only takes place after satisfying the below equations (1), (2), (3)

$$\left|p_o - q_o\right| < \alpha\left(Index_A\right) \tag{1}$$

$$\left|p_1 - p_o\right| < \beta\left(Index_B\right) \tag{2}$$

$$\left|q_1 - q_o\right| < \beta\left(Index_B\right) \tag{3}$$

The thresholds α and β are dependent on both Quantization Parameter (QP) and encoder selected offset values. The table index values $Index_A$, $Index_B$ are given by the following equations,

$$Index_A = Min\left(Max\left(0, QP + Offset_A\right), 51\right) \tag{4}$$

$$Index_B = Min\left(Max\left(0, QP + Offset_B\right), 51\right) \tag{5}$$

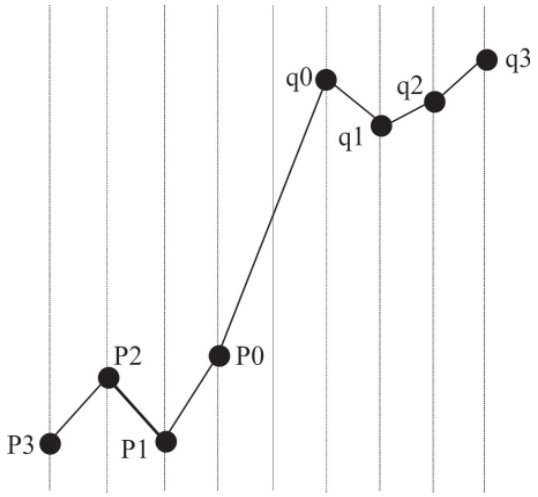


Figure 2: Condition where filtering is turned on

For luminance samples, the following additional spatial activities are checked to determine the extent of filtering,

$$\left|p_2 - p_o\right| < \beta\left(Index_{6B}\right) \tag{6}$$

$$\left|q_2 - q_o\right| < \beta\left(Index_B\right) \tag{7}$$

2.2 Filter operations

2.2.1 Filtering operations for Boundary strength value for Bs = 1 to 3

For boundary strength from 1 to 3, the value of p_0 and q_0 are modified as below

$$p_o^{'} = p_o + \Delta_o \; and \tag{8}$$

$$q_o^{'} = q_o - \Delta_o \tag{9}$$

The Δ_o value is calculated in two step process, first Δ_{oi} is calculated and the clipping is applied to this Δ_{oi} value

$$\Delta_{0i} = \left(4\left(q_0 - p_0\right) + \left(p_1 - q_1\right) + 4\right) \gg 3 \tag{10}$$

The values of p_1 and p_2 are modified, if the corresponding equations (6) and (7) are satisfied. The values are modified by the below equations

$$p_1^{'} = p_1 + \Delta_{p1} \tag{11}$$

$$q_1^{'} = q_1 + \Delta_{q1} \tag{12}$$

The Δ_{p1}, Δ_{q1} is calculated in two step process, first Δ_{p1i} is calculated and the clipping is applied to these Δ_{p1i} value

$$\Delta_{\perp} p1i = \left(p_{\perp}2 + \left(\left(p_{\perp}0 + q_{\perp}0 + 1\right) \gg 1\right) - 2p_{\perp}1\right) \gg 1\right) \tag{13}$$

The clipping process that has been applied to the Δ_{oi}, Δ_{p1i}, Δ_{q1i} are discussed below.

2.2.1.1 Clipping process

Clipping process is used to reduce the blurring resulted from too much low pass filtering. In clipping, a significant part of the intermediate values Δ_{oi}, Δ_{p1i}, Δ_{q1i} is limited in the range –c1 to c1. The c1 value is get from the 2-dimensional table that is indexed by $Index_A$ and Bs. For an increase in $Index_A$ and Bs value, the c1 value will keep increases providing a strong filtering

$$\Delta_{p1} = Min\left(Max\left(-c_1, \Delta_{p1_1}\right), c_1\right) \tag{14}$$

$$\Delta_{q1} = Min\left(Max\left(-c_1, \Delta_{q1_1}\right), c_1\right) \tag{15}$$

For clipping the delta value, the c0 is set to c1 first and for each true conditions of (6), (7) the c1 is incremented by 1.

$$\Delta_0 = Min\left(Max\left(-c_0, \Delta_{0i}\right), c_0\right) \tag{16}$$

In case of chrominance samples, the filtering is only applied to p0 and q0 values. For clipping the c0 value is initially set to c1 plus 1.

2.2.2 Filter operations for Boundary strength (Bs = 4)

In case of luminance filtering, for boundary strength equal to 4 a strong 4-tap and 5-tap filter or a weak 3-tap filter is applied based upon the sample value. The strong filter modifies up to three samples including edge sample on each side. The weak filter modifies only the edge sample. For applying the strong filter, the conditions in (17) has to be satisfied

$$\left|p_0 - q_0\right| \langle (\infty \rangle \rangle 2) + 2 \tag{17}$$

If both the conditions (6) and (17) are satisfied, the filtering is applied by the below equations

$$p_0^{'} = \left(p_2 + 2p_1 + 2p_0 + 2q_0 + q_1 + 4\right) \rangle \rangle 4 \tag{18}$$

$$p_1^{'} = \left(p_2 + p_1 + p_0 + q_0 + 2\right) \rangle \rangle 2 \tag{19}$$

$$p_2^{'} = \left(2p_3 + 3p_2 + p_1 + p_0 + q_0 + 4\right) \rangle \rangle 3 \tag{20}$$

In case of chrominance filtering, if either of the conditions (6) or (17) is satisfied then only p_0 is changed according to the following equations and p_1, p_2 are left unchanged

$$p_0^{'} = \left(2p_1 + p_0 + q_1 + 2\right) \rangle \rangle 2 \tag{23}$$

For modifying the q values, conditions (6) is replaced by (7) and the same filtering process is repeated by replacing p sample positions by q sample positions.

2.3. Deblocking filter in H264/SVC

The deblocking filter is used to remove the blocking artifacts produced due to motion compensation and quantization process. In H264/AVC the deblocking filter is applied for the reconstructed frame to remove the blocking artifacts results from motion compensation and quantization process. The H264/SVC consists of several layers from base layer to enhancement layer providing increased scalability in terms of spatial resolution, temporal resolution and quality. For H264/SVC the deblocking filter is applied in the same manner as H264/AVC, additionally the deblocking filter is applied in the interlayer prediction and a special consideration has to be done for MBAFF coded frames since it is widely supported in H.264/SVC. The deblocking filter operation is same for normal case and interlayer prediction, for the later, some additional condition has to be included in applying the deblocking filter. In the inter-layer prediction process of H264/SVC the enhancement layer data is predicted from previously reconstructed data of base layer. In case of inter-layer prediction, the deblocking filter is applied only for the I_BL type macroblock to the corresponding 4x4 co-located blocks. In I_BL type macroblock, all luma blocks of enhancement macroblock corresponds to lower resolution layer blocks of intra-picture coded. Since the deblocking filter consumes 30% of total computation time, an effective filtering in terms of faster computation is necessary to improve the efficiency of H264/SVC.

The interlaced type frames consist of top and fields which are captured at different time instants, the top field consist of odd number of rows and bottom field consist of even number of rows from the frame's initial position [19]. The frames are coded either using PAFF or MBAFF coding in H.264/SVC encoder. In PAFF, the two fields can either combined as single coded frame (frame mode) or coded as two separate fields (field mode) for a single frame. While in MBAFF coding, each vertical macroblock pair is coded either in field or frame mode. In frame mode, the macroblock pair contains the frame lines. In case of field mode, for each macroblock pair the top macro block contains top field lines and the bottom macroblock contains bottom field lines, doubling the spatial extent of the field coded macroblock. In H264/SVC, the MBAFF coding for interlaced frames is widely used. In deblocking filter operation, the filtering on the MB edges includes pixel from neighboring MB, creating dependency due to the coding type of neighboring macroblock. Since for normal PAFF coded frames an entire frame is either coded in field or frame, the above mentioned dependency is avoided for filtering operation. In MBAFF coding, the adaptability of field or frame mode is for each vertical macroblock pair, so a higher dependency is created on filtering the MB edge increasing the complexity of deblocking filter. Efficient deblocking filter architecture is needed to reduce the complexity and faster filtering for MBAFF coded frames.

3. Proposed Method

In this paper, a normal filtering architecture is designed for PAFF coded frames and parallel filtering architecture for MBAFF (frame/field mode for each macroblock pair) coded frames. The normal filtering architecture uses filtering unit pair which performs both horizontal and vertical filtering simultaneously. Since for filtering a 4x4 macroblock, the macroblock has to be filtered four times this requires repeated memory access. The proposed filtering architecture helps in reducing the number of memory access providing faster filtering operation. The filtering unit is capable of performing the above mentioned filtering operation. For PAFF coding, the choice between frame or field mode is applicable for entire frame. Normal filtering process consisting of single filtering unit pair is assigned

to current MB providing faster filtering. For MBAFF coded frames, a parallel filtering architecture with two filtering unit pair is assigned to the macroblock pair for faster and efficient filtering. The control unit in both Input and Output Buffer Controller Unit perform additional functionalities to store and retrieve the vertical macroblock pair data in proper order. The proposed method adapts for both normal filtering process and MBAFF filtering. In case of normal filtering process, the additional modules that are used in MBAFF coding are disabled by the method of clock gating. In case of inter-layer prediction, the filtering is only applied for I_BL type macroblock which is of intra-coded [13]. The filtering order is same for deblocking filter in inter-layer prediction process, but the Filtering Unit is designed to check whether the macroblock is of intra-coded I_BL type otherwise filtering is disabled for that particular macroblock.

3.1 Normal filtering operation

In case of normal filtering operation, a single filtering unit pair is assigned to a macroblock. The proposed architecture for deblocking filter in normal filtering process is given in Figure 3. The filtering pair consists of horizontal and vertical filters. The filtering operation is done subsequently for all edges in the row and column using corresponding horizontal and vertical filters. The vertical filtering for the horizontal edges takes place simultaneously except for first filtering operation which starts after two MB cycle. The filtering architecture consists of Input Buffer Controller, Filtering Module and Output Buffer Controller. The Filtering Module contains a Filtering Unit accompanied with Input Control, Output Control and a Transpose Unit. This organized filtering architecture helps in effective and faster filtering of each MB. Each Unit in the filtering architecture is given in detail below.

3.1.1 Input Buffer Controller

The input buffer controller unit consists of a control unit, two separate buffer unit each of it store a 4 x 4 data block of size (4x32) bit. In the two buffer unit, one is used to get data from reference memory (previous reconstructed MB) and other buffer unit is used to hold current MB data that has to be filtered. The reference memory and current MB data is loaded to both horizontal and vertical filtering unit. The buffer unit consists of a register array to store the macroblock for simpler data accessing. The control unit provides proper data accessing method from the buffer unit to each filtering unit. For horizontal filtering the data is accessed in the row order from the buffer unit, while for vertical filtering the data is accessed in the column order from the buffer unit. For MBAFF coded frames, proper macroblock from each vertical macroblock pair has to be accessed for parallel filtering process.

3.1.2 Filtering module

Filtering Module consist of Filtering Unit accompanied with sub units of Input Control, Output Control and a Transpose Unit. The sub units help in effective data movement to and from the Filtering Unit providing faster filter operation.

3.1.2.1 Filtering Unit

The Filtering unit is capable of performing the above mentioned adaptive filtering operation based upon the slice level, edge level and sample level. Since QP and offset values are same for 4x4 blocks. Each Filtering Unit computes the boundary strength and threshold values only once for the 4x4macroblock. The Filtering Unit is configured to perform both horizontal and vertical filtering.

3.1.2.2 Input Control

The Input Control helps in choosing the data given to the Filtering Unit. The Input Control consists of two buffers (FIFO). In those two buffers, one of it is used to store the data for current MB is of size (4x32) bit and the other is used to store the reference macroblock from Reference memory or Output Control Unit is of size (4x32) bit. This buffer helps in simultaneous data loading and filtering. In horizontal filtering, it chooses the data from reference memory (previous reconstructed data), Current MB data and transpose of previous filtered data. For vertical filtering, the Input Control additionally receives semi-filtered pixel from output buffer (FIFO buffer) which is stored temporarily in it.

3.1.2.3 Output Control with Transpose Unit

The Output Control with Transpose Unit consists of two temporary buffers (FIFO) and an additional buffer (FIFO) of size (4x32) bits each. It is employed in both Filtering Module to get filtered output 'p' and 'c' from the Filtering Unit and forwards it to the corresponding next stage. In horizontal filtering, the output control forwards it's either to the vertical filtering unit for filtered pixel 'r' or Input Control of same filtering unit for filtered pixel 'c' except for the last edge in the row in which both 'r', 'c' is forwarded to vertical filtering unit. For vertical filtering the output data is either forwarded to same Vertical Filtering Unit for filtered pixel 'c' or to the Output Buffer Unit (for future vertical filtering) for filtered pixel. The additional buffer stores the filtered pixel 'p' in case of last edge in the row. The Transpose Unit used in our filtering architecture is different from normal Transpose Unit used in [8]. Since the horizontally filtered pixel data is given to the Vertical Filtering Unit, the pixel data is transposed to convert the pixel

accessing order from row wise to column wise. After final vertical filtering of macroblock, the transpose unit converts the macroblock in to normal mode (i.e. for column wise to row wise), which is stored in Immediate Reference Memory for the next MB filtering, Reference Memory for subsequent next row filtering.

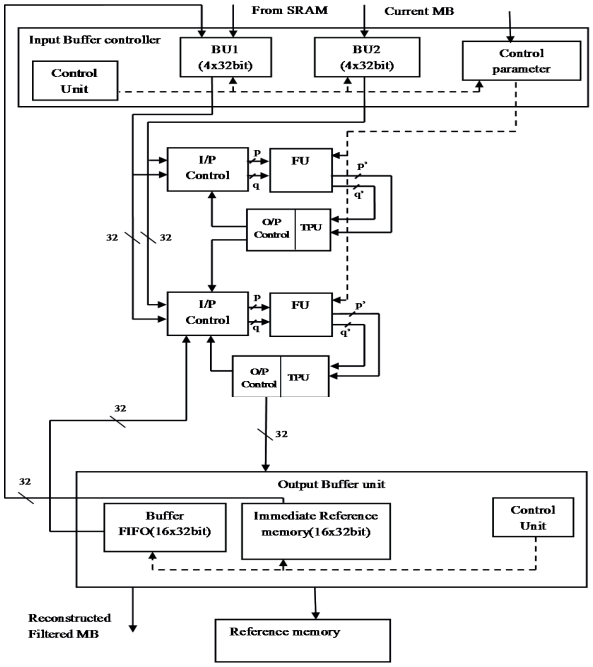


Figure 3: Normal filtering architecture

In the normal transpose module, the transposing operation for the pixel data is performed at the output, while in our filtering architecture the transposing operation is done at the input itself by the control unit. In case of normal transpose architecture, the transpose operation is applied after getting the 4x4 block of data, results in complexity in storing the future filtered output. Since in our proposed architecture, the transposing operation applied at input level helps in reducing the complexity in storing the future filtered pixel and accessing the transposed output. Figure 4. shows the data storing order in the buffer for subsequent horizontal and vertical filtering.

3.1.3 Output Buffer Controller Unit

The Output Buffer Controller Unit consists of several storage units such as Temporary Buffer (FIFO), Immediate Reference Memory and a Control Circuit to support the filtering operation. The Output Buffer Controller Unit receives the filtered data from the vertical filtering unit and by using control circuit the filtered pixel output is moved to the appropriate storage unit. The temporary buffer holds the semi-filtered data which has been later used for subsequent vertical filtering of the current MB. The Immediate Reference Memory (size

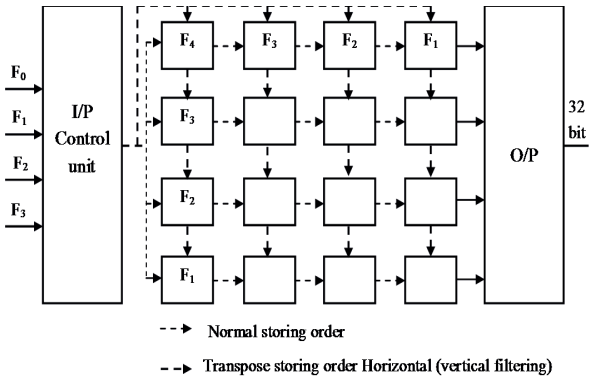


Figure 4: Normal filtering architecture

16x32bit) holds the last column of the final filtered MB as reference pixel data for filtering first vertical edge of the next MB. The control circuit bypasses the filtered pixel data to Reference memory for subsequent row filtering in the current Frame.

3.1.4 Memories

The Immediate Reference Memory holds the last column of filtered MB which consists of four 4x4 macroblock for subsequent macroblock filtering in the frame. The Buffer FIFO is also used to hold the four 4x4 semi-filtered macroblock for future filtering operations of same macroblock. The memory consumed by Immediate Reference memory and Buffer FIFO is of 1K, in which each occupies 512 bits. Since most FPGA has multiple SRAM slots and the dual port SRAM is used as Immediate Reference Memory and Buffer FIFO in our architecture.

3.1.5 Filter processing order

In normal filtering process, a filtering unit pair is used for simultaneous horizontal and vertical filtering. Initially vertical filtering starts after two horizontal filtering cycles based on the filtering order. Figure 5 shows the filtering order for the given macroblock. For filtering a 4x4 macroblock, 31 clock cycles is required. To filter a 16x16 luma MB, 121 clock cycles is needed and for two 4x4 chroma macroblock, 80 clock cycles is needed.

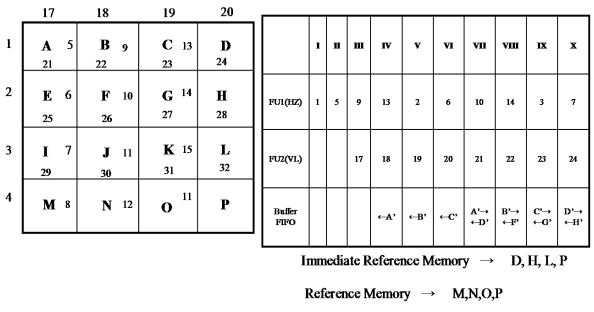


Figure 5: Filter processing order

So totally 201 clock cycles is required to filter a macroblock. To filter a HD frame of resolution 1920x1080, the number of clock cycles required to filter all the luma block is (8100 x 121) clock cycles and for all the chroma blocks is (8100 x 80) clock cycles. The edge filtered in each Filtering Unit and the semi-filtered data that are moved in and out from Buffer FIFO as given in Figure 5. After filtering a current MB, the filtered macroblock stored in Immediate Reference Memory are D, H, L, P and in Reference Memory are M, N, O, P.

3.2 MBAFF filtering operation

For MBAFF coded frames, the current and adjacent MB (reference MB) is coded either in frame or field mode. Thus, for filtering current MB edges combinations of frame/field, field/field, frame/frame and field/frame modes have to be considered. The proposed system provides a novel parallel filtering and data accessing order to reduce this complexity for efficient and faster filtering. In our proposed method, the filtering always takes place for both bottom and top field lines of the frame which requires novel macroblock accessing order from the current MB pair for both field/frame modes.

Meanwhile the macroblock of adjacent MB are always stored as field mode in reference memories for future reference, thus complexity in filtering due to the above mentioned dependency is greatly reduced. Based on the current MB mode, the adjacent macroblock are accessed in proper 4x4 blocks for filtering (i.e. directly for field mode or frame mode). In filtering vertical macroblock pair, each edge is represented by an 8x8 blocks, filtering have to takes place for these blocks. In the 8 x 8 blocks, the filtering for each 4 x 4 block is independent of each other, so an effective parallel filtering architecture provides a faster filtering of these 8x8 blocks. Two filtering unit pair is used for these parallel filtering of each edge. Figure.6 shows the filtering architecture for MBAFF coded frames. Each pair of filtering unit is assigned to the macroblock in the pair. Each filtering unit simultaneously filters the 4x4 block of the corresponding macroblock pair in horizontally and vertically. Initially in each Filtering Unit pair vertical filtering takes place after two horizontal filtering. Since the filtering unit consists of combinational circuit and in MBAFF coding the filtering for each edge takes place for 8x8 blocks, a large memory is needed to store the semi-filtered data and the reference data. These two filtering unit pairs help in achieving faster filtering of the macroblock pair. The Input Buffer Controller Unit consists of two Buffer Unit for storing the reference macroblock and current vertical macroblock pair. The Buffer Unit used to store the reference macroblock data is of size 2(4 x32) bit.

The Buffer Unit used to store the vertical macroblock pair is of size 2(4x32) bit. The control unit in the Input Buffer Controller accesses the proper 4x4 macroblock from current MB and adjacent MB Buffer Unit to the two filtering unit pair according to the proposed filtering architecture. The Output Buffer Controller Unit consists of two Temporary Buffer (FIFO) of size (16x32) bit, such that each Temporary Buffer (FIFO) is used to store the corresponding semi-filtered data of the macroblock in the pair for later use. Comparing to the normal filtering

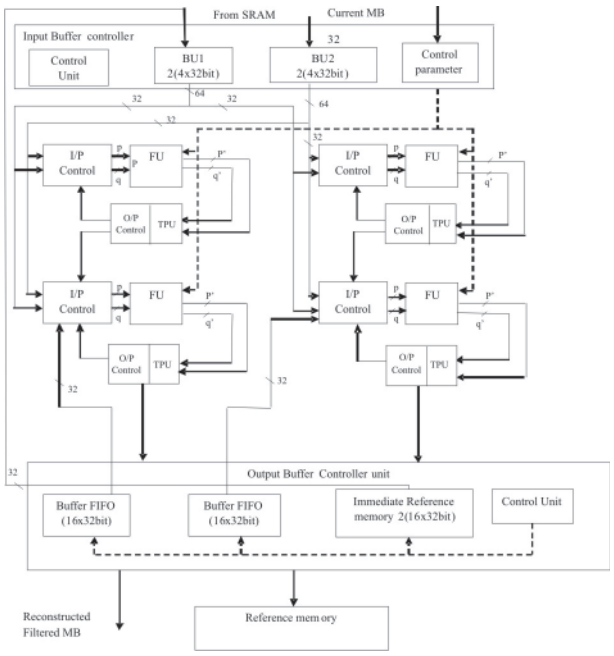


Figure 6: MBAFF filtering order

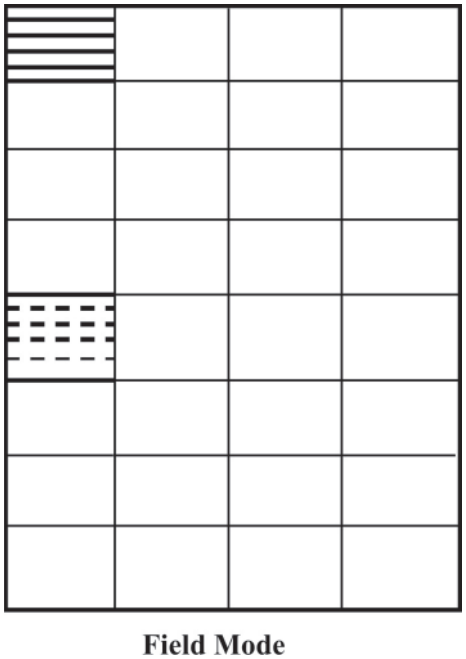


Figure 7: 4x4 macroblock accessing in Field mode

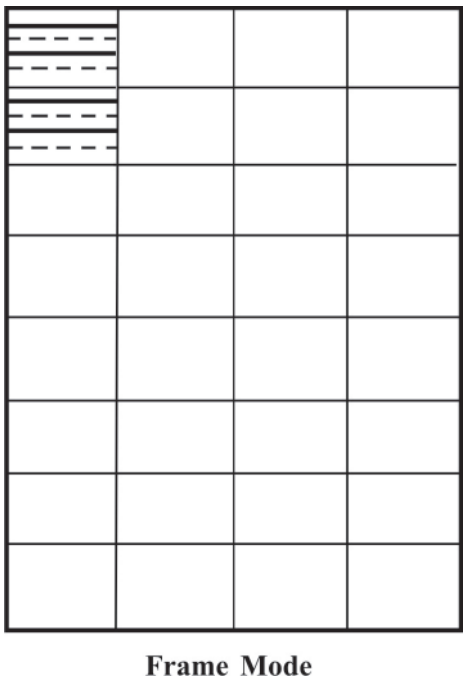


Figure 8: 4x4 macroblock accessing in Frame mode

process, the immediate reference memory size is also doubled to store the last column of the previous filtered macroblock pair. The Reference Memory size also consumes two times the memory used in normal filtering process. The filtered data is stored in unique manner in the reference memory to support the filtering architecture by means of faster accessing. Compared to normal filtering process, the filter architecture for MBAFF coding consumes twice its area but the speed has been improved.

3.2.1 Filtering method for macroblock

In our proposed method, each current 8 x 8 macroblock constitutes the field lines of both macroblock in the pair and the filtering will take place for both 4 x 4 top fields and 4 x 4 bottom fields.

Since each current MB may be either of field or frame mode, a proper accessing of field lines in the macroblock pair is required. In case of both first vertical and horizontal edge filtering, the current 8 x 8 macroblock is filtered with the previous filtered macroblock which is of field or frame mode. Thus proper filtering order and a reference data accessing order is required for efficient filtering. In case of current macroblock of field mode, parallel filtering is applied for the macroblock 1, 2 in the vertical macroblock pair as given in the Figure 7. The same filtering order is adopted for the whole vertical macroblock pair. For current macroblock of frame mode, the successive macroblock 1, 2 in the vertical MB pair is accessed for parallel filtering as given in Figure 8.

This filtering order is applied for whole MB pair in frame mode.

3.2.2 Data Storing in Reference memories

The proposed filtering architecture overrides the dependency between the current MB and reference macroblock due to different coding modes adopted in both macroblock (i.e. field or frame mode).To support this filtering architecture, the final filtered MB is stored in a field format in the Immediate Reference Memory and in the Reference Memory. The control unit in the Output Buffer Unit stores the previous filtered MB always in field mode, such that the reference macroblock can be accessed according to the current macroblock mode (i.e. either directly for field mode or frame mode). This helps in reducing the complexity in data accessing of reference macroblock due to the mode dependency. The order in which the filtered MB stored in reference Memories for frame and field mode is given in Figure 9.

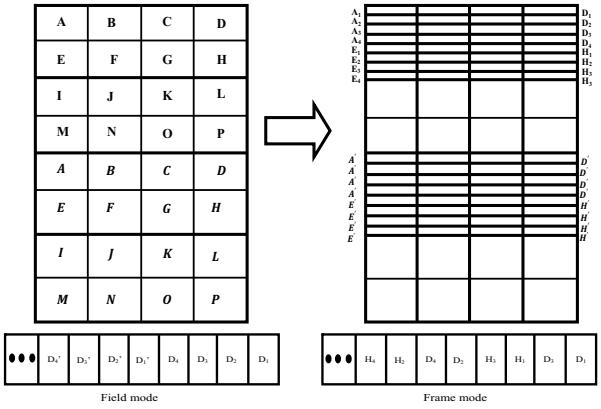
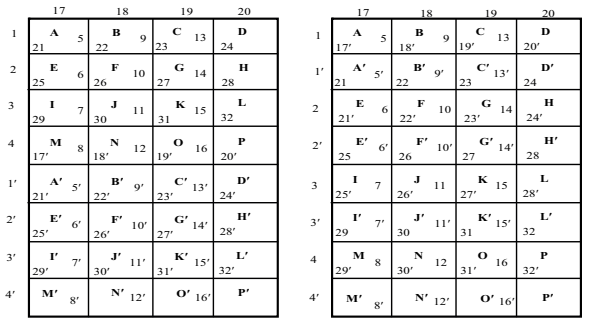


Figure 9: Data Storing order in the Immediate Reference Memory and Reference Memory

3.2.3 Filter processing order

Two filtering unit pair with individual Buffer FIFO is used for parallel filtering of macroblock pair. Each Filtering Unit pair works as the normal filtering operation processing the corresponding macroblock in the pair. Since in MBAFF filtering of macroblock pair works as two normal filtering processes, the number of clock cycles required to filter the vertical macroblock pair is same as the normal filtering process. Since for filtering a 4x4 macroblock 31 clock cycles is required. In filtering the vertical macroblock pair, for 16x16 luma MB 121 clock cycles is needed and for two 4x4 chroma macroblock 80 clock cycles is needed. Additionally some 20 cycles are required for MBAFF filtering. The total number of clock cycles required to filter a vertical macroblock pair is 221 clock cycles. To filter a HD frame of interlaced type, the number of clock cycles required to filter all the macroblock is (8100 x 221) clock cycles.



	I	II	III	IV	V	VI	VII	VIII	IX	X
FU1(HZ)	1	5	9	13	2	6	10	14	3	7
FU2(VL)			17	18	19	20	21	22	23	24
FU3(HZ)	1’	5’	9’	13’	2’	6’	10’	14’	3’	7’
FU4(VL)			17’	18’	19’	20’	21’	22’	23’	24’
Buffer FIFO1				←A	←B	←C	A→ ←D	B→ ←F	C→ ←G	D→ ←H
Buffer FIF01				←A’	←B’	←C’	A’→ ←D’	B’→ ←F’	C’→ ←G’	D’→ ←H’

Immediate Reference Memory D, D’, H, H’, L, L’P, P’
Reference Memory M, M’, N, N’, O, O’, P, P’

Figure 10: Filtering Order for MBAFF frames

The Figure 10 shows the filtering order of the macroblock pair in each Filtering Unit (FU) and the semi-filtered data which is moved in and out from Buffer FIFO. The filtered macroblock stored in Immediate Reference Memory is given by D,D’,H,H’,L,L’,P,P’ and in Reference Memory is given by M,M’,N,N’,O,O’,P,P’.

4 Results

The proposed deblocking filter for H264/SVC is implemented in Cyclone V (5CEFA9F31C8N) and the results are analyzed.

Table 2: Comparison of proposed filtering architecture with other filtering architecture

	Gate count	Processing Cycles per MB	Frequency (MHz)	Memory
[7]	19.64k	250	100	864+8N
[8]	24k	446	100	1000
[9]	20.66k	614	100	640
Proposed Normal	18.1k	202	200	3768
MBAFF	29k	242	200	7536

Compared to other filtering architecture, our proposed architecture achieves 19 % increase in processing speed. Since temporary buffer is used to store the semi-filtered pixel information, it helps in saving a significant number of clock cycles in accessing the semi-filtered pixel for further filtering process. In addition, the vertical macroblock pair is filtered in parallel and the adjacent MB is stored in field mode in the reference memories to avoid the dependency between the current and adjacent MB, which in turn reduces the complexity of deblocking filter. In transpose module, for the filtered output the proposed method applies transposing operation at the input level, helps in reducing the complexity in storing the future filtered pixel and accessing the transposed output. As a result, the proposed system achieves 30 % complexity reduction in the deblocking filter. Table 2 shows the comparison of deblocking filter with various architectures. Some additional clock cycles has been spent on proper accessing of proper macroblock in the pair which has been compensated by the reduction in complexity. Since the H264/SVC supports various level of layers with scalable resolution in terms of spatial, temporal and quality. This deblocking filter can be effectively implemented in various layers of different resolution by adopting the in-built SRAM slot for memories. The number of memory references for filtering a macroblock is also reduced. The proposed deblocking filter will filter the whole MB in 201 clock cycles for both luma and chroma blocks. In case of MBAFF filtering, the processing will takes place in 221 clock cycles. The proposed filter architecture occupies 8 % less area compared to other filtering architecture.

5 Conclusion

The deblocking filter operation for H264/SVC has more complexity compared to other operation. The filter has to be adaptable for PAFF/MBAFF coded frames and inter-layer prediction. A novel filtering order with parallel processing and efficient data accessing method is applied to the deblocking filter in H264/SVC for faster filtering. The proposed architecture has reduced memory references compared to other filtering architecture. The architecture is implemented in Cyclone V (5CEFA9F31C8N) and performance improvement in terms of processing speed (i.e. number of clock cycles for filtering) of 19 % and area reduction by 8 % is achieved.

6 References

1. Heiko Schwarz, Detlev Marpe, Member, IEEE, and Thomas Wiegand, Member, Overview of the Scalable Video Coding Extension of the H.264/AVC, IEEE transactions on circuits and systems for video technology, Vol. 17, No. 9, 2007.
2. Video Codec for Audiovisual Services at p x 64 kbit/s, ITU-T Rec. H.261, ITU-T, Version 1: (1990), Version 2: 1993.

3. Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbit/s—Part 2: Video, ISO/IEC 11172-2(MPEG-1 Video), ISO/IEC JTC 1, 1993.
4. Generic Coding of Moving Pictures and Associated Audio Information—Part 2: Video, ITU-T Rec. H.262 and ISO/IEC 13818-2(MPEG-2 Video), ITU-T and ISO/IEC JTC 1, 1994.
5. Thomas Wiegand, Gary Sullivan, Ajay Luthra, Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC), 2003.
6. P. List, A. Joch, J. Lainema, G. Bjntegaard, and M. Karczewicz, Adaptive deblocking filter, IEEE Trans. Circuits Syst. Video Technol.,Vol. 13, No.7, 2003, 614–619.
7. T. M. Liu, W. P. Lee, T. A. Lin, and C. Y. Lee, A memory-efficient deblocking filter for H.264/AVC video coding, Proc. IEEE ISCAS, Vol. 3, 2005, 2140-2143.
8. B. Sheng, W. Gao, and D. Wu, An Implemented architecture of deblocking filter for H.264/AVC, Proc. IEEE ICIP, Vol.1, 2004, 665-668.
9. Y. W. Huang, T. W. Chen, B.Y. Hsieh, T. C. Wang, T. H. Chang, and L. G. Chen, Architecture design for deblocking filter in H.264/JVT/AVC, Proc.IEEE ICME, 2003693-6.
10. T. Wiegand, G. J. Sullivan, J. Reichel, H. Schwarz, and M. Wien, Eds., Amendment 3 to ITU-T Rec. H.264 (2005) j ISO/IEC 14496-10:2005, Scalable Video Coding, 2007.
11. C. A. Chien, H. C. Chang, and J. I. Guo, A High Throughput In-Loop Deblocking Filter Supporting H.264/AVC BP/MP/HP Video Coding, Proc. IEEE APC-CAS, 2008,312-315.
12. C. A. Chien, H. C. Chang, and J. I. Guo, A High Throughput Deblocking Filter Design Supporting Multiple Video Coding Standards, Proc. IEEE ISCAS, 2009,2377-2380.
13. Andrew Segalland Gary J. Sullivan, Fellow, Spatial Scalability within the H.264/AVC Scalable Video Coding Extension C, IEEE transactions on circuits and systems for video technology, Vol. 17, No.9, 2007.
14. H. Huang, W. Peng, T. Chiang e H. Hang, Advances in the Scalable Amendment of H.264/AVC, Communications Magazine, IEEE. Vol. 45, No.1, 2007, 68-76.
15. Rijkse, K, Video coding for low bit rate communication, ITU-T Recommend H.263, Vol. 34, No. 12, 1998, 42-45.
16. S.Wenger, H.264/AVC over IP, IEEE Trans. Circuits Syst., Vol. 13, No.7, 2003, 645–656.
17. Secker and D. Taubman, Motion-compensated highly scalable video compression using an adaptive 3D wavelet transform based on lifting, in Proc. ICIP, Vol. 2, 2001, 1029–1032.
18. T. M. Liu, W. P. Lee, and C. Y. Lee, An in/post-loop deblocking filter with hybrid filtering schedule, IEEE Trans. Circuits Syst. Video Technol., Vol. 17, No.7, 2007, 937-943.
19. E. Francois, J. Viéron, and V. Bottreau, Interlaced coding in SVC, IEEE Trans. Circuits Syst. Video Technol., Vol. 17, No.9, 2007, 1136–1148.
20. H. Schwarz, T. Hinz, D. Marpe, and T. Wiegand, Further Progress on Scalable Extension of H.264, ITU-T SG 16/Q 6 (VCEG), Doc. VCEGX08, 2004.
21. C. M. Chen and C. H. Chen, An efficent architecture for deblocking filter in H.264/AVC video coding, in Proc. IASTED Int. Conf. Comput.Graphics Imaging, 2005, 177–181.
22. T. Cervero1, A. Otero2, S. López1, E. De La Torre2, G. Callicó1, R. Sarmiento1, T. Riesgo2, A Novel Scalable Deblocking Filter Architecture for H.264/AVCand SVC Video Codecs, Multimedia and Expo (ICME), 2011 IEEE International Conference, 2011.

Arrived: 18. 09. 2014
Accepted: 06. 01. 2015