A pipelined VLSI architecture for Sample Adaptive Offset (SAO) filter and deblocking filter of HEVC

Sha Shen, Weiwei Shen, Yibo Fan[a], and Xiaoyang Zeng
State Key Lab of ASIC and System, Fudan University,
825 Zhangheng Road, Shanghai, 201203, China
a) *fanyibo@fudan.edu.cn*

Abstract: This paper present a high throughput design for Sample Adaptive offset (SAO) filter and deblocking filter used in an HEVC decoder. A five-stage pipelined architecture is proposed to support both SAO filter and deblocking filter on a 32×32 pixel block basis. Deblocking filter and SAO filter can work simultaneously in consecutive pipeline stages. The on-chip SRAM can also be shared by deblocking filter and SAO filter. Coupled with the novel filter order, an interlaced SRAM memory mapping scheme is proposed to increase the throughput for deblocking filter. The experimental results show that our design can support $4K \times 2K$@60 fps (4096×2304) HEVC video sequence at the working frequency of only 60.8 MHz.

Keywords: HEVC, Sample Adaptive Offset (SAO) filter, deblocking filter, in-loop filter
Classification: Integrated circuits

References

[1] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm and T. Wiegand: ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 Document JCTVC-K1003 (2012).
[2] C. Fu, E. Alshina, A. Alshin, Y. Huang, C. Chen, C. Tsai, C. Hsu, S. Lei, J. Park and W. Han: IEEE Trans. Circuits Syst. Video Technol. **22** [12] (2012) 1755.
[3] A. Norkin, G. Bjontegaard, A. Fuldseth, M. Narroschke, M. Ikeda, K. Andersson, M. Zhou and G. V. Auwer: IEEE Trans. Circuits Syst. Video Technol. **22** [12] (2012) 1746.
[4] D. Zhou, J. Zhou, J. Zhu and S. Goto: IEICE Trans. Fundamentals **E92-A** [12] (2009) 3203.
[5] W. Shen, Y. Fan and X. Zeng: IEICE Trans. Electron. **E95-C** [4] (2012) 441.

© IEICE 2013
DOI: 10.1587/elex.10.20130272
Received April 15, 2013
Accepted April 19, 2013
Publicized May 22, 2013
Copyedited June 10, 2013

1 Introduction

High Efficient Video Coding (HEVC) [1] is the emerging new video coding standard, which is under the development by the Joint Collaborative Team on Video Coding (JCT-VC). HEVC can reduce 50% bit rate without compromising the picture quality in comparison with H.264/AVC [2].

Deblocking filter is used as the in-loop filter of H.264 to suppress the block artifacts introduced by prediction or quantization error. A similar deblocking filter is also used in HEVC. But the computational complexity of the deblocking filter is reduced. Only the 8×8 block boundaries which are also prediction unit boundaries or transform unit boundaries are filtered in HEVC [3].

Various new coding tools are used in HEVC in order to further improve the coding efficiency. The transform size of 2D DCT used in HEVC can be up to 32×32, which is much larger than the transform size of H.264. Larger size transform can bring more blocking artifacts or ringing artifacts. The number of interpolation taps used for the inter prediction of HEVC is also higher than that of H.264 [2]. The ringing artifact is further exaggerated for HEVC.

In order to suppress the artifacts mentioned above, a new filter named Sample Adaptive Offset (SAO) is applied along with deblocking filter for HEVC. A block diagram of HEVC decoder is shown in Fig. 1. The blocks inside the reconstruction loop are marked in yellow color while other blocks are in green color. SAO is located between the reference picture buffer and the deblocking filter. The in-loop filter of HEVC consists of two filters: SAO filter and deblocking filter. Deblocking filter is applied to the reconstructed picture first. SAO filter is then applied to whole picture which is fully filtered by deblocking filter.

In this paper, a five-stage pipelined VLSI architecture is proposed for SAO filter and deblocking filter of HEVC. SAO filter and deblocking filter can work simultaneously due to the pipelined architecture. This proposed

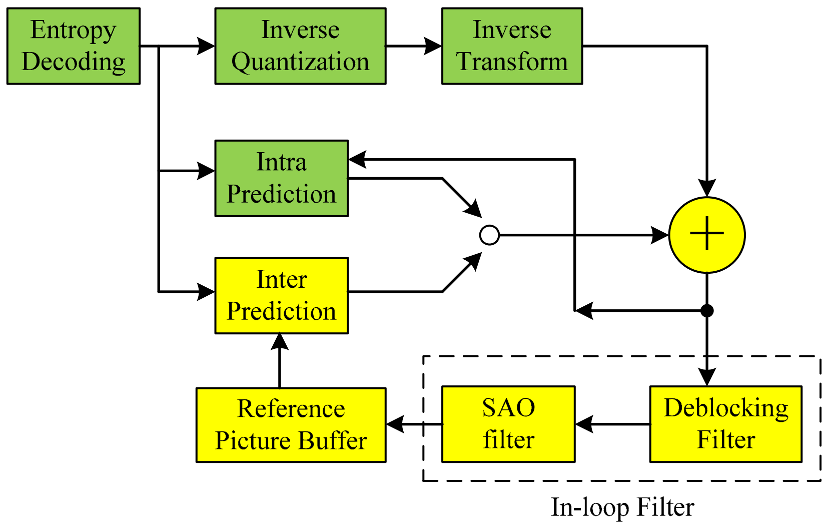

Fig. 1. Block diagram of an HEVC decoder

© IEICE 2013
DOI: 10.1587/elex.10.20130272
Received April 15, 2013
Accepted April 19, 2013
Publicized May 22, 2013
Copyedited June 10, 2013

design can processes one 32 × 32 pixel block in 110 clock cycles. On-chip SRAM can be shared by both SAO filter and deblocking filter, which greatly reduces the hardware cost. Coupled with the novel filter order, an interlaced SRAM memory mapping scheme is proposed to increase the throughput for deblocking filter. As a result, this design can support 4K × 2K@60 fps (4096 × 2304) HEVC video sequence at the working frequency of only 60.8 MHz.

2 In-loop filter algorithm of HEVC

Two different filters named SAO filter and deblocking filter are used in HEVC. Deblocking filter is applied the whole picture first. After all the pixels in the picture have been filtered by the deblocking filter, SAO filter is applied to get the final picture which can be used as a reference picture for inter prediction. The algorithm of SAO filter and deblocking filter is introduced as below.

The processing flow of deblocking filter is shown in Fig. 2 (a). Boundary decision is performed first. Only the 8 × 8 block boundary is filtered by deblocking filter. The 8 × 8 block boundary will be filtered only when this boundary is also a boundary of coding unit, prediction unit or transform unit. Boundary strength (BS) reflects how strong the filtering is applied to the boundary. The value of BS is an integer ranging from 0 to 2. BS is determined by some coding information such as prediction mode, motion vector (MV) and so on. Threshold values β, tc used for filter on/off decision, strong/weak filter selection are derived based on the QP of P block and Q block in Fig. 2 (b). As shown in Fig. 2 (b), P block and Q block are two adjacent 4 × 4 blocks across the boundary involved in filtering. The value d is also involved in the filter decision of filter on/off, strong and weak filter. This parameter is derived from the value of twelve pixels in the first and the fourth line. These twelve pixels are labeled as red circles shown in Fig. 2 (b).

These four lines involved in filtering share some common decisions including β, tc, d. Each line also has its own parameters such as dE, dEp and dEq, which are used for the decision of filter on/off, strong and weak filter for each line.

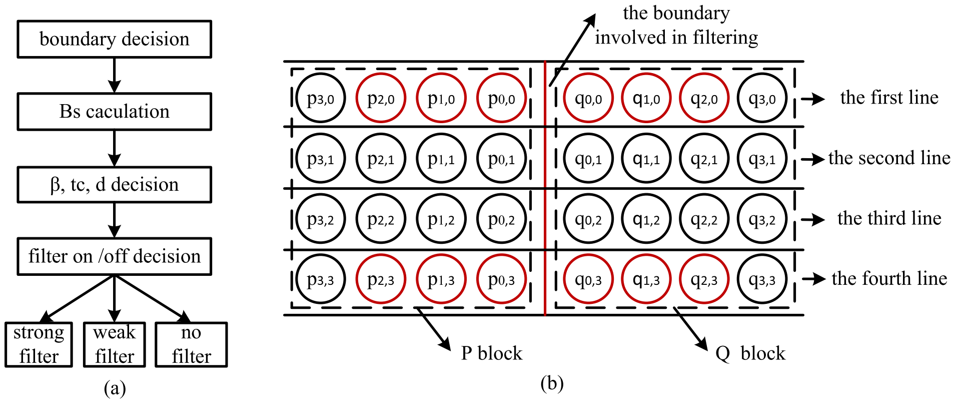
boundary decision
Bs caculation
β, tc, d decision
filter on /off decision
strong filter
weak filter
no filter
(a)
the boundary involved in filtering
p3,0 p2,0 p1,0 p0,0 q0,0 q1,0 q2,0 q3,0
p3,1 p2,1 p1,1 p0,1 q0,1 q1,1 q2,1 q3,1
p3,2 p2,2 p1,2 p0,2 q0,2 q1,2 q2,2 q3,2
p3,3 p2,3 p1,3 p0,3 q0,3 q1,3 q2,3 q3,3
the first line
the second line
the third line
the fourth line
P block
Q block
(b)

Fig. 2. Algorithm of HEVC deblocking filter: (a) Overall deblocking flow. (b) Filter decision for two adjacent 4 × 4 blocks

© IEICE 2013
DOI: 10.1587/elex.10.20130272
Received April 15, 2013
Accepted April 19, 2013
Publicized May 22, 2013
Copyedited June 10, 2013

$$\mathrm{dE} \neq 0, \ \mathrm{bs} \neq 0 \tag{1}$$

$$\mathrm{bs} > 1 \tag{2}$$

The values of eight pixels across the boundary are donated as $p_{3,0}$, $p_{2,0}$, $p_{1,0}$, $p_{0,0}$, $q_{0,0}$, $q_{1,0}$, $q_{2,0}$, $q_{3,0}$, which are labeled as the first line in Fig. 2 (b). The luminance component of the first pixel line is filtered only when the condition of equation (1) is satisfied. The chrominance component of the first pixel line is filtered only when the condition of equation (2) is satisfied. The decisions of strong and weak filter are detailed in [1].

Two types of SAO filters are used in HEVC: Edge Offset (EO) and Band Offset (BO). Each coding tree unit (CTU) can only select one type of SAO filter and the whole CTU is then filtered by either edge offset filter or band offset filter. When edge offset filter is applied, an offset is added to each pixel in the CTU according to the values of both the current pixel value and its neighboring pixels. Four different approaches can be used to select the two neighboring pixels for edge filter among eight neighboring pixels: horizontal, vertical, 135 diagonal and 45 diagonal. As shown in Fig. 3, the current pixel is marked as "c" and the two neighboring pixels are marked as "a"/"b".

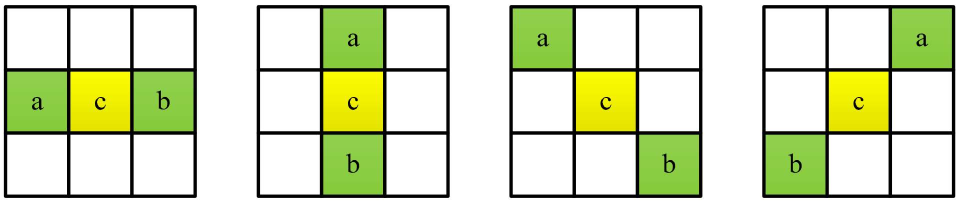

Fig. 3. Four types of neighboring pixels for edge offset filter

When band offset filter is applied, an offset is added to each pixel in the CTU according to the current pixel value. The pixel value range is divided into 32 bands and each band has its own offset. But only four offsets (not all 32 offsets) are signaled to the decoder side [2] and other 28 offsets are set as zero.

3 Proposed VLSI architecture

The top level diagram of the proposed in-loop filter for an HEVC decoder is shown in Fig. 4. This design is divided into 5 pipeline stages to maximize the throughput. Five two-port SRAM blocks (SRAM_C0, SRAM_C1, SRAM_L0, SRAM_L1 and SRAM_T) are used to store the pixels of current coding unit as well as the left/top neighboring pixels for the deblocking filter. Horizontal line buffer and vertical buffer are used to store the neighboring pixels used by SAO filters. They are marked as "V/H line buffers" in Fig. 4. Two horizontal pixel lines (marked as brown dots in Fig. 6) from top-side pixel block are stored in horizontal line buffer. Two vertical pixel lines (marked as green dots in Fig. 6) from left-side pixel block are stored in vertical line buffer.

In order to speed up the pixel loading process from external memory, a

© IEICE 2013
DOI: 10.1587/elex.10.20130272
Received April 15, 2013
Accepted April 19, 2013
Publicized May 22, 2013
Copyedited June 10, 2013

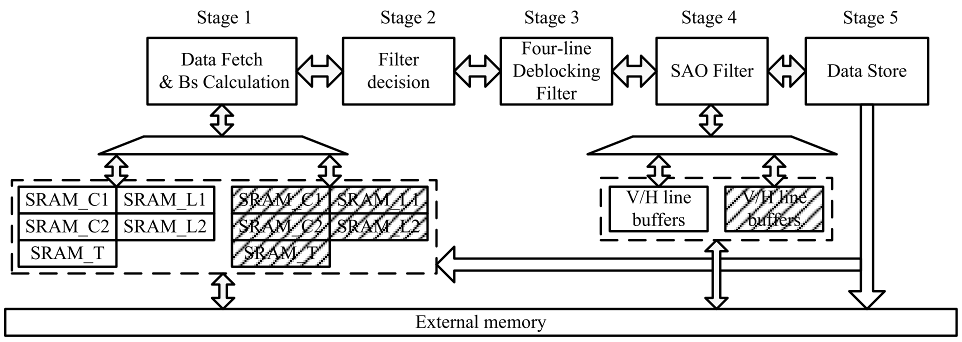

Fig. 4. Top level diagram of the HEVC in-loop filter

double-buffering scheme is shown in Fig. 4. When the data in current SRAM blocks and line buffers are being processed, the SRAM blocks and line buffers marked with shadow lines are loading the pixels for next processing.

3.1 Introduction of the 5-stage pipelined architecture

A 5-stage pipelined architecture is shown in Fig. 4. The computational complexity of deblocking filter is much higher than SAO filter. So the first three stages are used for deblocking filter. SAO filter is located in the 4th stage. The function of each pipeline stage is described as following:

Stage 1: Boundary strength information is calculated in this stage. SRAM access signals are also generated to fetch the pixels out from SRAM blocks for the in-loop filtering.

Stage 2: Various parameters for deblocking filter such as the threshold value β, tc, the condition dE, dEp, dEq are calculated. These parameters will be used for the decision of filter on/off, strong or weak filter.

Stage 3: The boundary over two adjacent 4×4 blocks is filtered by deblocking filter according to the parameters Bs, tc, dE, dEp, dEq obtained in previous stage. The exact filtering algorithm is described in [1].

Stage 4: In this stage, SAO filter is applied to the two adjacent 4×4 blocks which have been filtered by the deblocking filter. Parts of the pixels in horizontal or vertical line buffer are also filtered by SAO filter.

Stage 5: The filtered pixels are stored back to on-chip memories or external reference picture buffer.

The pixels for next processing are loaded from external memory and stored into the SRAM blocks (SRAM_T, SRAM_C0 and SRAM_C1) or horizontal/vertical line buffers which are marked with shadow lines in Fig. 4. The SRAM blocks SRAM_L0 and SRAML1 are updated by the 5th pipeline stage as shown in Fig. 4.

The timing diagram of the proposed in-loop filter is shown in Fig. 5. SAO filter (in pipeline stage 4) is disabled when the deblocking filter is processing the vertical edges. After 51 cycles, all 48 vertical edges are processed. SAO filter is enabled from cycle 52 to 107. The labels of each box in Fig. 5 are the two 4×4 blocks which are been processed at current clock cycle. For example, the label "L0C0" at cycle 1 means that pipeline stage 1 is processing the pixels from 4×4 blocks L0 and C0. Blocks L0/C0 are shown in Fig. 6 and

© IEICE 2013
DOI: 10.1587/elex.10.20130272
Received April 15, 2013
Accepted April 19, 2013
Publicized May 22, 2013
Copyedited June 10, 2013

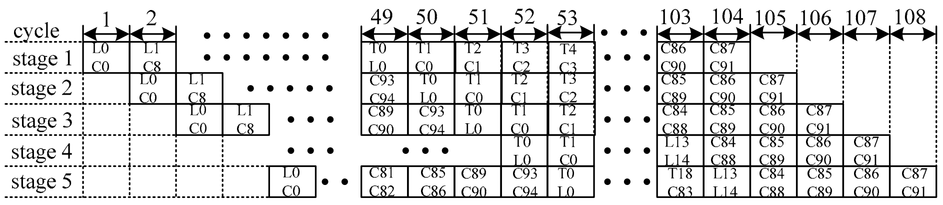

Fig. 5. Timing diagram of the HEVC in-loop filter

explained in Section 3.3. 104 clock cycles are needed by each pipeline stage to process the whole 32 × 32 pixel block, which is explained in section 3.3. The latency introduced by the 5-stage pipeline architecture is 4 cycles. The control logic need 2 additional cycles to enable the in-loop filter core at the beginning and disable the in-loop filter core at the end. The whole 32 × 32 pixel block can be processed in 110 (104 + 4 + 2) clock cycles.

3.2 Proposed filter order scheme for deblocking filter

The vertical edges in a picture are filtered first. The vertical edges are filtered from left to right in their geometrical order. After all the vertical edges inside current 32 × 32 pixel block have been filtered by the deblocking filter, the horizontal edges are filtered from top to bottom. The samples used for horizontal edge filtering are the modified samples of vertical edge filtering.

The boundaries involved in filtering on a 32 × 32 pixel block basis are labeled as red lines in Fig. 6. The vertical boundaries (if available) including {v1, v2, v3, v4, v5, v6, v7, v8} are filtered from left to right. The horizontal boundaries (if available) including {h1, h2, h3, h4, h5, h6, h7, h8} are filtered from top to bottom.

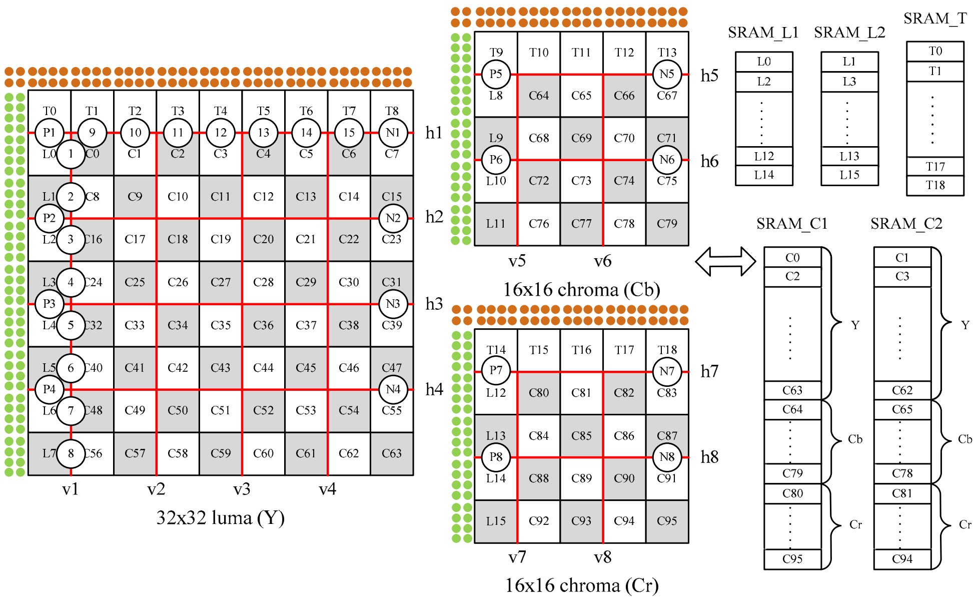

Fig. 6. Interlaced memory mapping scheme for on-chip SRAM

© IEICE 2013
DOI: 10.1587/elex.10.20130272
Received April 15, 2013
Accepted April 19, 2013
Publicized May 22, 2013
Copyedited June 10, 2013

Each boundary consists of several four-line units. The vertical boundaries are filtered from top four-line unit to the bottom four-line unit (e.g., from four-line unit 1 to four-line unit 8 in vertical boundary v1). The horizontal boundaries are filtered from left four-line unit to the right four-line unit (e.g., from four-line unit 9 to four-line unit 15 in horizontal boundary h1). Each four-line units can be filtered by the four-line deblocking filter core in one clock cycle.

The four-line units {P1, P2, P3, P4, P5, P6, P7, P8} are available when the left boundary (v1) of current 32×32 pixel block is not the left boundary of the current picture frame. These four-line units should be filtered if they are available. Otherwise they should be skipped.

But the four-line units {N1, N2, N3, N4, N5, N6, N7, N8} are not filtered unless the right boundary of current 32×32 pixel block is also the right boundary of current picture frame. The four-line units {N1, N2, N3, N4, N5, N6, N7, N8} in current 32×32 pixel block will be the four-line units {P1, P2, P3, P4, P5, P6, P7, P8} in the right-side 32×32 pixel block.

3.3 Interlaced memory mapping for deblocking filter

The on-chip memory is used to reduce the I/O bandwidth between the chip and the system. The main challenge is to properly arrange the data in memory modules so that the pixels can be smoothly fetched out from the on-chip memory and sent to the four-line filter core. The direction of block boundary can be either vertical or horizontal. We present an interlacing memory organization to access the data effectively in the processing of both vertical and horizontal filtering.

Our approach is to divide the on-chip memory into five modules (SRAM_C1, SRAM_C2, SRAM_L1, SRAM_L2, and SRAM_T), as is shown in Fig. 6. SRAM_C1 and SRAM_C2 store the pixels of the current 32×32 pixel block. Pixels in SRAM_L1 and SRAM_L2 come from the left neighboring 32×32 pixel block. Pixels in SRAM_T come from the top neighboring 32×32 pixel block and left-top neighboring 32×32 pixel block. Each square in Fig. 6 stands for a 4×4 pixels of luminance or chrominance component. The grey squares from C0 to C95 are stored into SRAM_C1. The white squares from C1 to C94 are stored into SRAM_C2. Other squares are also stored into SRAM_L1, SRAM_L2 or SRAM_T, which is shown in Fig. 6. Two adjacent 4×4 pixel blocks across the boundary to be filtered always come from different memory modules. As a result, all pixels can be easily accessed from the SRAMs on both the vertical and horizontal filtering operations.

A two-port SRAM can support one read access and one write access in the same clock cycle. Five two-port SRAM blocks are used in this proposed design. These SRAMs can also be used to store the immediate pixels for the further filter.

The filtering begins with the vertical boundary of L0 and C0, which is filtered by the four-line filter core. The filtered data of vertical edge will be stored back to the on-chip memory for horizontal boundary filter, which is shown in Fig. 6. The filtered data after horizontal filtering operation will

© IEICE 2013
DOI: 10.1587/elex.10.20130272
Received April 15, 2013
Accepted April 19, 2013
Publicized May 22, 2013
Copyedited June 10, 2013

be sent to SAO filter for further filtering. There are totally 104 edges to be filtered in a 32×32 pixel block. Each edge can be processed in one single clock cycle. So the deblocking filter core can process the whole 32×32 pixel block in 104 cycles.

3.4 SAO filter with horizontal/vertical line buffers

According to the filtering order of deblocking filter in this design, the vertical 8×8 block boundaries are filtered first. SAO filter is turned off for vertical 8×8 block boundaries. After all the vertical boundaries are filtered, the horizontal 8×8 block boundaries will be filtered and the output be further filtered by the SAO filter. In Fig. 7 (a), the 32 pixels (inside the solid line box) inside two 4×4 adjacent blocks are the outputs of deblocking filter and they are shown as blue dots with shadow line. The pixels in horizontal line buffer are shown as brown solid dots and the pixels in vertical line buffer are shown as green solid dots. Both horizontal and vertical line buffer are implemented with registers instead of SRAM. Four purple pixels with grid line are also implemented with registers and serve as a temporal buffer inside SAO filter.

SAO filter can be categorized into two types: edge offset or band offset. When band offset is applied, no neighboring pixels are needed. Each pixel is filtered according to its own value. But when edge offset is applied, each pixel is filtered according to the values of both itself and its neighboring pixels which are stored in vertical line buffer or horizontal line buffer. Only 32 pixels out of all 60 pixels in Fig. 7 (a) can be filtered by the SAO filter. These 32 pixels are marked with dash-line in Fig. 7 (a).

The algorithm of SAO filter is quite straightforward, as is descripted in [1]. In order to filter one pixel by SAO edge offset filter, only two comparators are needed to compare the current pixel value with the values of two neighboring pixels and one offset is added to current pixel value according

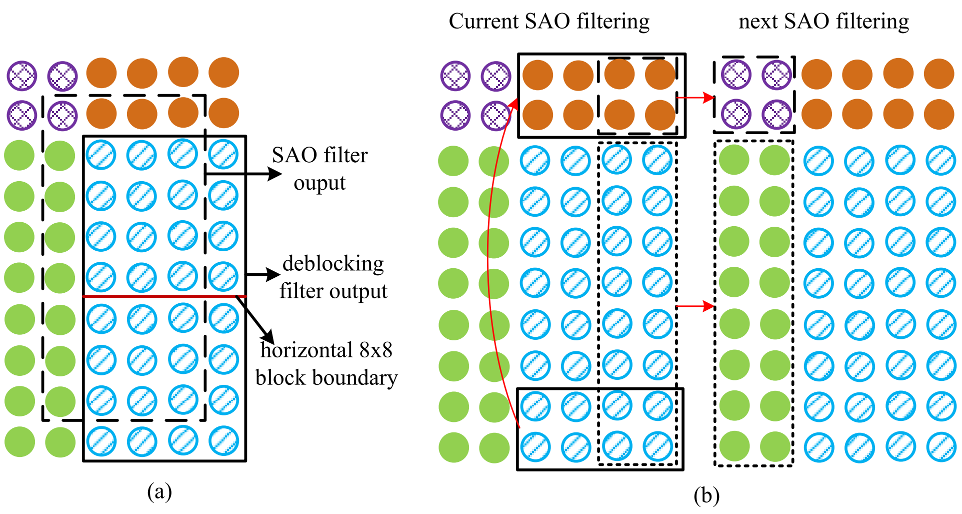

Fig. 7. SAO filter for two adjacent 4×4 blocks: (a) 32 filtered pixels by SAO filter. (b) Horizontal/vertical line buffer updating

© IEICE 2013
DOI: 10.1587/elex.10.20130272
Received April 15, 2013
Accepted April 19, 2013
Publicized May 22, 2013
Copyedited June 10, 2013

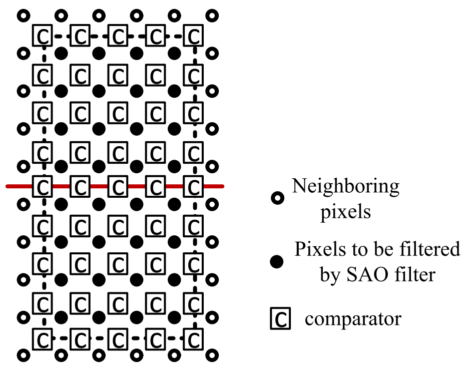

Fig. 8. Comparators shared by neighboring pixels

to the comparison. In case of SAO band offset filter, the comparison with neighboring pixels can be skipped and an offset is added to current pixel value according to its own value. The comparators can also be shared by neighboring pixel. 32 solid dots in Fig. 8 are the pixels which will be filtered by SAO filter. If 2 comparators are used for each pixel, 64 comparators are required for the proposed SAO filter in Fig. 7. The proposed comparator sharing technique is shown in Fig. 8. Every comparator has 4 neighboring pixels and 2 neighboring pixels are chosen as the inputs of current comparator according to the type of edge offset. In order to support all the four types of edge offset SAO filter, 5 comparators are needed in one row direction and 9 comparators are needed in column direction. Totally 45 (5×9) comparators and 32 adders are needed in order to filter the 32 pixels (marked by dash line in Fig. 7 (a) and Fig. 8) in one clock cycle.

After the SAO filtering for two adjacent 4×4 blocks is done, the pixels in horizontal/vertical line buffer need to be updated. Please note that the pixels stored in horizontal/vertical line buffer are the outputs of deblocking filter, not the outputs of SAO filter. The updating process is shown in Fig. 7 (b). The blue pixels inside dot line are moved into the vertical line buffer. 4 pixels marked with purple grid line are also updated by the pixels marked inside the dash line box in Fig. 7 (b). Then the blue pixels inside solid line will be stored into the horizontal line buffer, which is shown in Fig. 7 (b). These 8 pixels will be used for the 4×4 block pair in next row. These 8 pixels are not used for the filtering process of the 4×4 block pair which is right to current 4×4 block pair. For example, if T0 and L0 in Fig. 6 are the current 4×4 block pair which is filtered by the SAO filter, the 8 pixels above T0 (marked as 8 brown dots in Fig. 6) are updated by the 8 pixels in L0 after T0/L0 have been filtered. These 8 pixels are used when L1 and L2 are being filtered. After T0 and L0 are filtered, T1/C0 will be the next 4×4 block pair to be filtered. The 8 pixels above T1 (another 8 brown dots in Fig. 6) will be used for next SAO filtering. When all the buffers are updated, the SAO filtering for another two adjacent 4×4 blocks in horizontal direction can proceed.

4 Implementation results

The proposed architecture is implemented in Verilog-HDL and synthesized

© IEICE 2013
DOI: 10.1587/elex.10.20130272
Received April 15, 2013
Accepted April 19, 2013
Publicized May 22, 2013
Copyedited June 10, 2013

Table I. Gate count summary

Block name	Gate count(NAND2)
Four-line debloking filter	21K
SAO filter	24K
Data load/store (including bs caculation)	10K
Horizontal/vertical line buffer	9K
SRAM	44K
Total	108K

under a timing constraint of 61 MHz with 0.13-um ASIC standard cell library. The gate count of each module is summarized in Table I.

According to Table I, the gate count of proposed in-loop filter core for HEVC is 45K $(21 + 24)$. As shown in Fig. 6, there are 164 pixels (1312 bits) are stored in the horizontal line buffer and 152 pixels (1216 bits) are stored in vertical line buffer. The total size of SRAM used in this design is 16768-bit. The gate count of the SRAM and line buffers is totally 53K, nearly 50% of the silicon area. If the double-buffering scheme is used, the size of SRAM and line buffers should be doubled. The experimental result demonstrates that the proposed architecture achieves a trade-off between high-throughput performance and memory size. In addition, the hardware cost of the control logic is 10K in term of gate count.

The working frequency needed to support a specific video sequence can be calculated by equation (3). Here W is picture width and H is the picture height. Format is set as 1 for 4 : 2 : 0 YUV format and 2 for 4 : 4 : 4 YUV format. Fps is the frame rate. In order to support the target resolution (4096×2304@60 fps and 4 : 2 : 0 format), the working frequency is 60.8 MHz according to equation (3).

$$\text{Freq} = \frac{\text{W} * \text{H} * \text{Format} * \text{Fps}}{32 * 32} * 110 \tag{3}$$

The comparison with previous works is listed in Table II. Please note that the previous works in Table II only support H.264 deblocking filter while this work support the in-loop filter of latest HEVC standards. Both deblocking filter and SAO filter are used in the HEVC in-loop filter. But SAO filter is not supported by H.264. So the gate count of this proposed design shown in

Table II. Comparison with previous works

Reference	[4]	[5]	Proposed
Year	2009	2012	2013
Gate count(K)	30.2	30.6	31(1)
Technology	0.13um	0.13um	0.13um
Application Target	QFHD@60fps (3840x2160)	4Kx2K@30fps (4096x2304)	4Kx2K@60fps (4096x2304)
Frequency(MHz)	93.3	70.8	60.8

Note: (1) The gate count here is only for deblocking filter, SAO filter, SRAM and line buffer are not included.

© IEICE 2013
DOI: 10.1587/elex.10.20130272
Received April 15, 2013
Accepted April 19, 2013
Publicized May 22, 2013
Copyedited June 10, 2013

Table II is only for deblocking filter. Our design can support 4K × 2K@60 fps video sequence at the lowest working frequency, which is only 60.8 MHz. Due to our pipelined architecture, the maximum working frequency of this proposed design can reach 160 MHz at the cost of slightly larger silicon area. This is also the first paper which proposes a high throughput VLSI architecture for SAO filter. So the comparison for SAO filter is not listed in Table II.

5 Conclusion

To the authors' knowledge, this paper firstly reports a hardware implementation on the in-loop filter of HEVC, which features high throughput. With the five-stage balanced pipeline and the proposed memory interlacing technique, this architecture can accomplish a 32×32 pixel block in 110 clock cycles. As a result, our design can support 4K × 2K 60 fps video sequence with merely 60.8 MHz working frequency. It is capable for higher resolution or low power applications.

Acknowledgments

This paper is supported by National High Technology Research and Development Program (863, 2012AA012001), State Key Lab of ASIC & System Project (11MS004), Specialized Research Fund for thr Doctoral Program of Higher Education (SRFDP, 20120071120021).

© IEICE 2013
DOI: 10.1587/elex.10.20130272
Received April 15, 2013
Accepted April 19, 2013
Publicized May 22, 2013
Copyedited June 10, 2013