A Novel High Performance Architecture for H.264/AVC Deblocking Filtering

Sebastián López, Félix Tobajas, Gustavo M. Callicó, Pedro A. Pérez,
Valentín de Armas, José F. López, and Roberto Sarmiento

ABSTRACT—This letter presents an architecture based on a new double-filter strategy to perform the adaptive in-loop filtering process specified by the H.264/AVC standard. The proposed architecture shows considerable advantages, both in terms of hardware cost and latency, when compared with the approaches found in the most recent literature.

Keywords—H.264/AVC, video decoder, deblocking filter, video architecture, CMOS.

I. Introduction

The H.264/AVC standard represents the state-of-the-art in video coding standards. It provides superior features compared with its predecessors, such as MPEG-2, MPEG-4, and H.263, in terms of coding efficiency and network friendliness. These improved characteristics are due to the application of several new coding tools within the compression process defined by the standard, such as variable block-size motion estimation, quarter-pixel motion compensation, intra prediction, and in-loop deblocking filtering.

In particular, the adaptive deblocking filter defined by the H.264/AVC standard greatly contributes to reducing the presence of annoying blocking artefacts in the decoded video sequence, providing up to an average of 10% transmission bit-rate savings [1]. However, these favorable features are obtained through a highly adaptive and computationally intensive process stated by the H.264/AVC standard, which represents a significant percentage of the whole processing effort carried out by a standard-based decoder [2], [3]. These demanding characteristics become even more stringent for high definition video applications, where high temporal rates and large frame dimensions are usual, making the design of efficient deblocking filtering architectures a critical issue in order to achieve real-time performance.

Manuscript received Dec. 01, 2006; revised Jan. 11, 2007.
This research was funded by the Spanish Government under project TEC2005-08138.
Sebastián López (phone: + 34 928 457335, email: seblopez@iuma.ulpgc.es), Félix Tobajas (email: tobajas@iuma.ulpgc.es), Gustavo M. Callicó (email: gustavo@iuma.ulpgc.es), Pedro A. Pérez (email: palexi@iuma.ulpgc.es), Valentín de Armas (email: armas@iuma.ulpgc.es), José F. López (email: lopez@iuma.ulpgc.es), and Roberto Sarmiento (email: roberto@iuma.ulpgc.es) are with Department of Electronic Engineering, University of Las Palmas de Gran Canaria, Las Palmas, Spain.

II. H.264/AVC Deblocking Filter

The H.264/AVC standard defines a one-dimensional filtering to for each 4×4 block edge of the reconstructed macroblocks (MBs). This process consists of a horizontal filtering of the vertical edges and a vertical filtering of the horizontal edges, as shown in Fig. 1.

The filtering process starts computing the boundary strength, which is a number ranging from 0 (no filtering) to 4. The boundary strength varies adaptively per block according to the coding conditions and the MB pixel values. For strength 4, up to three pixels on either side of the edge can be modified while for strengths between 1 and 3, only up to two pixels on either side of the edge may be affected. Once the strength has been determined for an edge, a gradient-like analysis is performed to determine if the sharp edge should be preserved or filtered to attenuate blocking artefacts. In this sense, the filter is disabled, regardless of the filter strength, when $|p0-q0|\geq\alpha$, $|p1-p0|\geq\beta$, or $|q1-q0|\geq\beta$, where α and β mainly rely on the quantization parameter.

This process is repeated with all the 4×4 blocks composing an MB (see Fig. 2) together with their top (T) and left (L) neighbors.

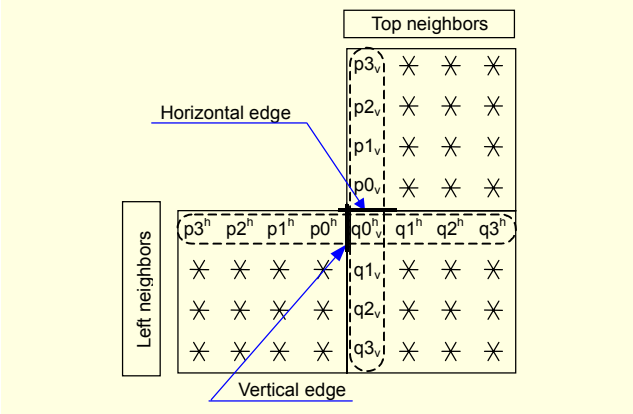

Fig. 1. Pixel distribution for horizontal and vertical filtering processes.

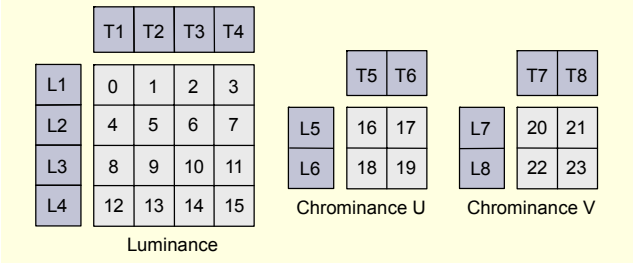

Fig. 2. Labeling of the 4×4 blocks of an MB and neighbors.

III. Proposed Architecture

Typical architectures presented for H.264/AVC deblocking filtering are based on a single filter unit for horizontal and vertical filtering [4]. To increase the filtering performance, we propose a novel deblocking filter architecture based on two identical filtering units, allowing simultaneous on-the-fly filtering of vertical and horizontal edges. Figure 3 shows the general block diagram of the proposed architecture in which all the data paths are 32-bits (4 pixels) wide.

The architecture first requests the top neighbors of the external memory via DMA (blocks T1 to T8) and the motion vectors to start the deblocking filtering process when all the top neighbors and motion vectors have been received. Figure 4 summarizes this filtering process within the proposed architecture in terms of block cycles, which are defined as 4 clock cycles (time consumed to process each 4×4 block).

The processing begins with the horizontal filtering of the left neighbor L1 (P input) and the current zero block (Q input), performed by the H filter unit. As the MBs are processed in lexicographical order, the right columns (blocks 3, 7, 11, 15, 17, 19, 21, and 23) are stored in the RAM neighbor module to be used as left neighbors (L1 to L8) in the filtering of the next MB. The RAM neighbor module comprises two 32×32 two-port (1R/1W) SRAM blocks, one for storing the left

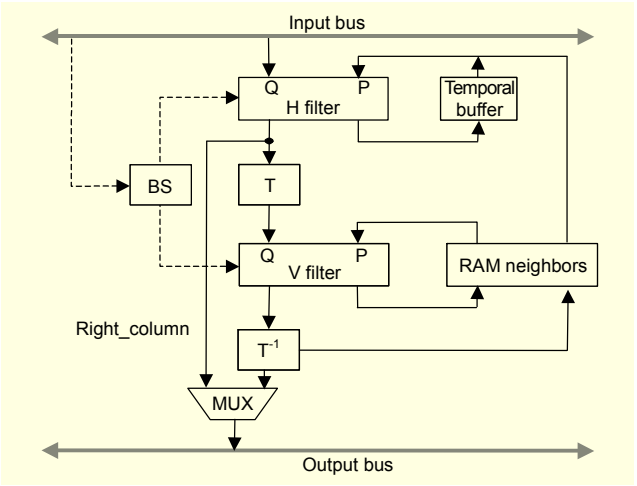

Fig. 3. Block diagram of the proposed architecture.

neighbors and the other for storing the top ones. The horizontal filtering starts as soon as the pixels from the current MB are loaded from external memory. This avoids the need to store all the data words internally at the input port before the filtering operation starts. A small temporal buffer module (4×4 pixels) is used to store the blocks after they are partially filtered in the horizontal direction. For the vertical direction, the RAM neighbor module is used instead. After horizontal filtering, the T unit starts transposing data from rows to columns. It then forwards transposed blocks to the V filter unit for the vertical filtering process. The BS unit computes the boundary strengths and the filter parameters required for both filtering directions. For MBs in the left or top borders of the frame, the boundary strength is set to zero to prevent the erroneous filtering of the pixels located at the borders of such MBs. After the luminance pixels are processed, the chrominance pixels are similarly processed. Finally, the first three columns of 4×4 blocks of the current MB (0, 1, 2, 4, 5, 6, and so on), the top neighbors (T1, T2, and so on), and the right column of the previous MB (3, 7, 11, 15, and so on) are transferred to the external bulk memory. These last blocks have been saved in the temporal buffer and come from the right column data path after the horizontal filtering, as shown in Fig. 3. This storage is performed concurrently with the loading of pixels from the next 4×4 top blocks and the reception of the updated configuration information and motion vectors for the next MB.

For the proposed architecture, the filtering process consumes 104 cycles, with four additional cycles required to transfer the last transposed block, and two cycles to register at the output of the filter units, resulting in a total count of 110 cycles to filter an MB.

IV. Results

The proposed architecture is described at the RTL level. It uses Verilog HDL. It has been tested exhaustively by

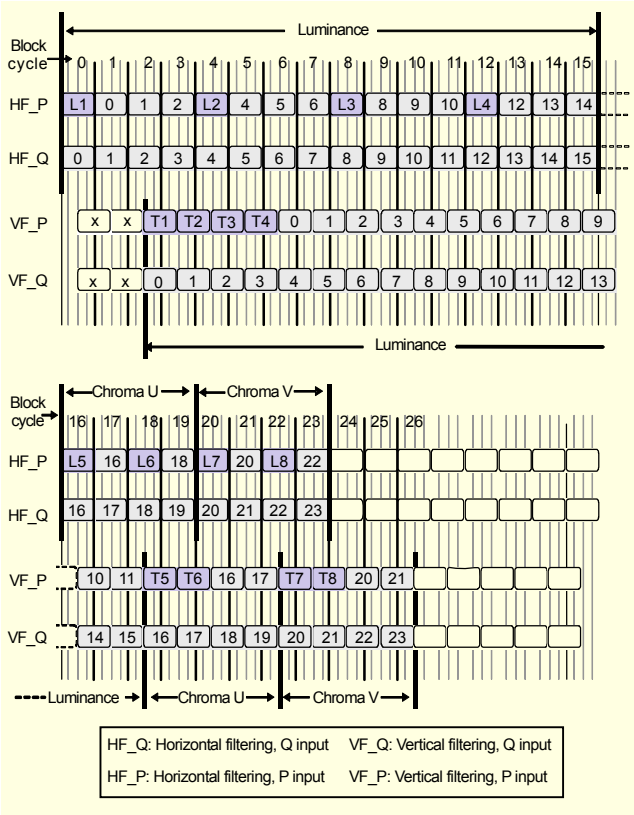

Fig. 4. Chronogram of the proposed architecture.

Table 1. Comparison with previously published works.

	Process (μm)	Filtering cycles / MB	SRAM requirements for pixels (bits)[1)]	Equivalent gates count
[5]	0.18	192	(128+1.5×FW) ×32	20.9 k
[6]	0.18	336	80×32	9.16 k[2)]
[7]	0.18	342	96×32	11.8 k[2)]
[8]	0.18	192	160×32	9.57 k[2)]
[9]	0.18	236	(160+2×FW) ×32	14.5 k
[10]	0.18	250	(96+2×FW) ×32	19.64 k
Prop.	0.18	110	64×32	12.60 k

1) FW represents the frame width in pixels
2) Gate count w/o boundary strength computation

comparing the results against several test patterns generated by the H.264/AVC standard C code [4]. The proposed architecture was synthesized at a frequency of 100 MHz with 0.18 μm CMOS standard cell technology from UMC (1.8 V). The results obtained in terms of equivalent gates count and the most significant architectural features of the proposed deblocking filter architecture are shown in Table 1.

Table 1 also shows the intrinsic characteristics of the state-of-the-art designs. This table demonstrates that the proposed architecture outperforms previous approaches. It takes less clock cycles to filter an MB with lower on-chip memory requirements because edges are filtered on-the-fly; thus, no input buffer is required. This is within a negligible increase in the final area of the architecture, since [6], [7], and [8] do not include the logic for boundary strength computation.

V. Conclusion

This letter outlined a novel architecture for the in-loop deblocking filter defined by the H.264/AVC standard. The proposed architecture significantly reduces the filtering latency incurred by previous architectures. The proposed architecture is able to process an HDTV-1080p (1920×1080 pixels @ 30 fps) video sequence at a frequency as low as 36.45 MHz without an increase in the design hardware cost as compared to the most recently published architectures.

References

[1] P. List, A. Joch, J. Lainema, G. Bjøntegaard, and M. Karczewicz, "Adaptive Deblocking Filter," *IEEE Trans. on Circuits and Systems for Video Techn.*, vol. 13, no. 7, July 2003, pp. 614-619.

[2] S.M. Park, M. Lee, S. Kim, K.-S. Shin, I. Kim, H. Cho, H. Jung, and D. Lee, "VLSI Implementation of H.264 Video Decoder for Mobile Multimedia Application," *ETRI J.*, vol. 28, no. 4, Aug. 2006, pp. 525-528.

[3] J.H. Han, M.Y. Lee, Y. Bae, and H. Cho, "Application Specific Processor Design for H.264 Decoder with a Configurable Embedded Processor," *ETRI J.*, vol. 27, no. 5, Oct. 2005, pp. 491-496.

[4] H.264/AVC Reference Software, http://iphome.hhi.de/suehring/tml/index.htm.

[5] S.-Y. Shih, C.-R. Chang, and Y.-L. Lin, "A Near Optimal Deblocking Filter for H.264 Advanced Video Coding," *Asia and South Pacific Conf. on Design Automation*, 2006, pp. 170-175.

[6] C.-C. Cheng and T.-S. Chang, "An Hardware Efficient Deblocking Filter for H.264/AVC," *IEEE Int. Conf. on Consumer Electronics*, 2005, pp. 235-236.

[7] S.-C. Chang, W.-H. Peng, S.-H. Wang, and T. Chiang, "A Platform Based Bus-Interleaved Architecture for Deblocking Filter in H.264/MPEG-4 AVC," *IEEE Trans. on Consumer Electronics*, vol. 51, no. 1, 2005, pp. 249-255.

[8] L. Li, S. Goto, and T. Ikenaga, "A Highly Parallel Architecture for Deblocking Filter in H.264/AVC," *IEICE Trans. on Information and Systems*, vol. E88-D, no. 7, 2005, pp. 1623-1629.

[9] G. Zheng and L. Yu, "An Efficient Architecture Design for Deblocking Loop Filter," *Picture Coding Symposium*, 2004.

[10] T.-M. Liu, W.-P. Lee, T.-A. Lin, and C.-Y. Lee, "A Memory-Efficient Deblocking Filter for H.264/AVC Video Coding," *IEEE Int. Symp. on Circuits and Systems*, vol. 3, 2005, pp. 2140-2143.