INTEGRATION, the VLSI journal 49 (2015) 65–77

Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

journal homepage: www.elsevier.com/locate/vlsi

A high performance 5 stage pipeline architecture for the H.264/AVC deblocking filter

N. Kefalas, G. Theodoridis *

VLSI Design Laboratory, Electrical and Computer Engineering Department, University of Patras, 26500 Patras, Greece

ARTICLE INFO

Article history:
Received 8 August 2012
Received in revised form
8 October 2014
Accepted 8 October 2014
Available online 23 October 2014

Keywords:
Deblocking filter
H.264AVC
Video compression
Pipeline
ASIC
FPGA

ABSTRACT

Exploiting specific properties of the algorithm, a high-throughput pipelined architecture is introduced to implement the H.264/AVC deblocking filter. The architecture was synthesized in 0.18 μm technology and the clock frequency and area are 400 MHz and 16.8 Kgates, respectively. Also, it is able to filter 217 and 55 Frames per second (Fps) for Full- and Ultra-HD videos, respectively. The introduced architecture outperforms similar ones in terms of frequency (1.8 × up to 4 ×), throughput, (1.5 × up to 3.8 ×), and Fps. Moreover, extensions to support different sample bit-depths and chroma formats are included. Also, experimental results for different FPGA families are offered.

1. Introduction

H.264/AVC (also called MPEG4 Part 10) developed by ITU-T and ISO/IEC MPEG [1] is the current industry standard for video coding. It is used in many commercial electronic devices and applications such as cellular phones, digital camcorders, high definition TV, and video teleconference systems. H264/AVC offers a remarkable improved performance in terms of video quality, bit-rate saving, and transmission resilience than the previous video coding standards like H.263, MPEG-2, and MPEG-4 [2]. Compared to MPEG-2, it achieves the same video quality improving the video compression by at least two times [3]. Also, it achieves up to 30% better compression than H.263 and MPEG-4 Advanced Simple Profile [4]. This encoding efficiency is achieved by incorporating a number of coding tools and algorithms such as multi-mode intra-prediction, context adaptive binary arithmetic coding (CABAC), and multi-frame and variable block-size inter-prediction with different pixel accuracy [1].

However, due to the incorporated block-based algorithms (e.g. Discrete Cosine Transform and Motion Compensation), the reconstructed video frame suffers from blocking artifacts and disturbing discontinuities. To overcome these problems, H.264/AVC includes an in-loop adaptive deblocking filter [5], which improves both the objective and subjective visual quality, whereas it also reduces the bit-rate requirements by 5–10% [2]. However, the use of the deblocking filter has a cost, which is its high computational complexity.

Firstly, the deblocking filter algorithm is highly adaptive and three filtering modes (strong, weak, and no filtering) may be applied. In particular, the algorithm is adaptive in three levels (slice, edge, and sample level), which results in a complex procedure with many conditions that must be checked to determine the applied filtering mode. Secondly, it is applied to each edge of all the 4 × 4 luma and chroma Sub-Blocks (SBs) of a 16 × 16 Macro-Block (MB), while each SB may be filtered up to four times. Third, it may update up to 3 pixels in each of the two adjacent SBs in each direction that the filtering takes place. Fourth, to determine the filtering mode and filter an edge of a SB, the involved pixels of the current and neighboring SBs must be read from the memory, which results in high memory-bandwidth requirements. Due to the above, about one-third of the total decoding time is spent on the deblocking filter making it a major bottleneck of the whole decoding process [5,6].

Thus, to achieve real-time decoding, the deblocking filter is usually implemented in hardware and a lot of architectures have been proposed for this purpose. These architectures are classified in three categories, which are (i) the single-filter non-pipelined architectures, [7–11], (ii) the single-filter pipelined architectures [12–16] and (iii) the architectures that include two (or more) filter units [17–22]. Comparing the above categories, the throughput is improved when the pipeline technique is applied and the number of the filter units increases with a cost in the area and complexity of the final design. Concerning the architectures of the second category to which the proposed one belongs, they are usually 4- or

* Corresponding author.
E-mail address: theodor@ece.upatras.gr (G. Theodoridis).

http://dx.doi.org/10.1016/j.vlsi.2014.10.002
0167-9260/

5-staged pipelined architectures characterized by advanced filtering orders of the MBs edges and pipeline organizations. Also, extra effort is paid to reduce the accesses of the external memory and the size of the on-chip memories.

In this paper, a 5-stage pipelined architecture is proposed to implement the deblocking filter algorithm. Analyzing the equations of the filter and exploiting specific properties of the algorithm, a novel architecture is introduced, which is characterized by (i) improved filtering order of the MBs' edges, (ii) pre-computation of the *BS*, c_0, α, and β parameters and threshold values by overlapping their calculations with the filtering operations, (iii) efficient distribution of the filter operations in the pipeline stages, (iv) highly-optimized design of the filter unit, (v) high data reuse, (vi) single-port on-chip memories, and (vii) coding information and upper neighbor MBs on on-chip memory. In the typical case, the introduced architecture needs 226 clock cycles to filter a MB.

The proposed architecture was implemented (for comparison reasons) in 0.18 μm CMOS technology and the achieved frequency and throughput are 400 MHz and 1770×10^3 MBs/s, respectively. Also, it is able to filter 217 Full-HD and 55 Ultra-HD Frames per second (Fps). In addition, extensions to support higher sample bit-depths (10, 12, 14 Bps) and different chroma sub-sampling formats (4:2:2 and 4:4:4), which are required by the H.264/AVC high profiles (Hi10P, Hi422P, Hi444P), are presented. The high optimization of the architecture allows the implementation of these profiles without significant impact on frequency, area, and throughput. Moreover, experimental results for different FPGA technologies are provided. Although, high routing overhead is occurred in the FPGA implementations, the presented architecture also achieves high frequency and throughput values.

Compared to similar single-filter pipelined architectures implemented in the same technology, the proposed architecture exhibits the smallest area. Also, it outperforms them in terms of frequency (1.8× up to 4×), throughput (1.5× up to 3.8×) and throughput/area (2.0× up to 7.4×) as well as in Fps (1.5× up to 3.7×) and (1.5× up to 3.9×) for Full- and Ultra-HD videos, respectively. Moreover, it is the only one among the existing single-filter pipelined architectures that achieves more than 50 Fps for Ultra-HD frames meeting the requirements of modern applications and avoiding the extra area that is required by the architectures which include more than one filter units.

Also, the proposed architecture has been compared with architectures that perform 2-D filtering. Specifically, these architectures include two filter units to filter both the vertical and horizontal edges concurrently. Although, the introduced architecture performs 1-D filtering, it is the best in terms of throughput and throughput/factors in most of the cases. Finally, comparing the presented architecture to the non-pipelined ones, the throughput is improved drastically, as it was expected, without significant area penalty. Specifically, the proposed architecture achieves the highest throughput/area value and outperforms the existing ones from 3.2× up to 5.4×.

A preliminary version of this work was presented in [24], which has been extended covering additional topics, which are: (i) the pipelined design of the filter unit, (ii) extensions of the initial architecture to support higher sample bit-depths and different chroma sub-sampling formats, which are required by the H.264 high profiles, along with the corresponding implementation results, (iii) FPGA implementation results and discussion, (iv) comparisons with single-filter non-pipelined architectures and pipelined architectures that include two filter units, and (v) study of the memory accesses and comparisons with the JM reference software implementation. Also, the pipeline organization and its features are presented in depth.

The paper is organized as follows. Section 2 presents the deblocking algorithm. In Section 3 the proposed architecture is described, whereas implementation results and comparisons are presented in Section 4. Finally, conclusions are provided in Section 5.

2. Deblocking filter algorithm

The deblocking filtering takes place on MB basis. As it is shown in Fig. 1, for each luma MB, firstly the vertical edges are filtered rightwards (from the edge V_0 to V_3) and then the horizontal ones downwards (from the edge H_0 to H_3).

To filter a sub-edge, 8 pixels ((q_0-q_3) and (p_0-p_3)) of two adjacent SBs are read from the memory and enter the filter unit, where (q_0, q_1, q_2, q_3) and (p_0, p_1, p_2, p_3) are the pixels of the current and neighboring (left or top) SBs, respectively. Also, to filter the edge V_0 (H_0), the pixels of the left (top) neighboring MBs are used. The same process is applied for the chroma components. Finally, before moving to the next MB, the filtering of the luma and chroma components of the current MB in both vertical and horizontal directions must be completed.

Fig. 2, which illustrates the deblocking filter algorithm for the luma sub-edges, shows clearly its adaptive nature [1,5]. There are several conditions (called flags) that must be checked to determine (i) if the sub-edge will be filtered or not and (ii) the applied filtering mode. To achieve this, a set of parameters and threshold values are used. The most important of them are the *Boundary Strength* (*BS*) and c_0 parameters and the α and β threshold values. Using them and the values of the pixels of the involved SBs, a number of flags are computed.

For instance, the *Filter Sample Flag* (*FSF*), which determines if the sub-edge will be filtered or not is calculated by the following equation:

$$FSF = (BS \neq 0)\ \&\&\ (|p_0 - q_0| < \alpha)\ \&\&\ (|p_1 - p_0| < \beta)\ \&\&\ (|q_1 - q_0| < \beta) \quad (1)$$

where && denotes logical AND operation, and α and β are threshold values derived from tables specified in [1] with a specific procedure described below.

Also, the *Strong Filter Flag P* (*SFF_P*), *Weak Filter Flag P* (*WFF_P*), and the *Strong Filter Flag Q* (*SFF_Q*) and *Weak Filter Flag Q* (*WFF_Q*) must be computed. Each of them is a function of the *BS* parameter and the α and β threshold values and they are used to determine the applied filtering mode. For instance, if *SFF_P*=1, the (p_0, p_1, p_2, p_3) pixels (*p*-pixels) are filtered applying a strong filter, whereas if *WFF_P*=1, a weak filter is used. More details can be found in [1].

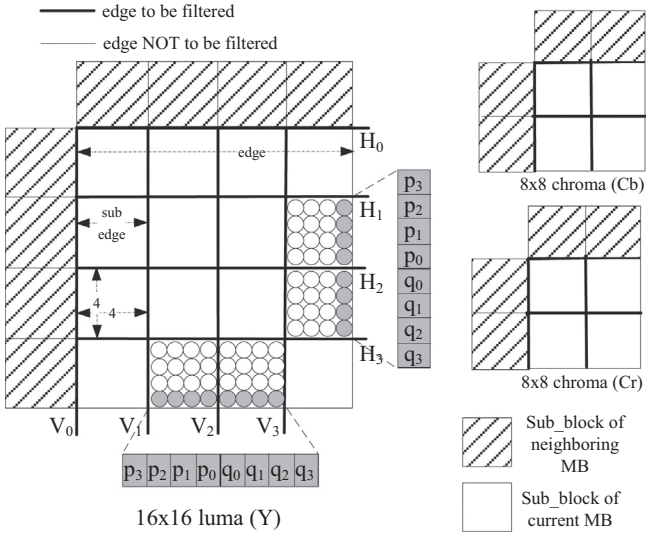

Fig. 1. Horizontal and vertical filtering of a 16 × 16 MB.