A Low-Power Deblocking Filter Architecture for H.264 Advanced Video Coding

Jaemoon Kim, Sangkown Na and Chong-Min Kyung
Department of EECS, KAIST
373-1, Yuseong-gu, Guseong-dong
Daejeon, Republic of Korea
Email: {jmkim, skna}@vslab.kaist.ac.kr, kyung@ee.kaist.ac.kr

Abstract—In this paper, a low-power deblocking filter architecture for H.264/AVC is proposed. A hybrid filtering order has been adopted to boost the speed of the deblocking filter process up to 208 clock cycles per 16x16 macroblock. The processing order of the filter is optimized to reduce power consumption and filter size and this is done by reducing memory access and raising the reusability of register blocks. A hardware implementation, under Samsung 0.18 μm standard cell library, consumes 18.34K gates at a clock frequency of 125MHz. Comparing to some state-of-the-art designs, the proposed architecture delivers the lowest level of power consumption while achieving similar speed of performance.

I. INTRODUCTION

H.264/AVC is an emerging coding standard [1] [2]. The H.264/AVC offers high coding efficiency in comparison to previous video standards like MPEG-2 or MPEG-4. The coding efficiency is gained from heterogeneous video coding standard algorithms such as incorporation of the inter/intra predictions, which use smaller block size (4x4 block), integer discrete cosine transform (DCT), context-adaptive variable length coding (CA VLC), and deblocking filtering. H.264/AVC is a block-based video coding standard algorithm, in which video is encoded and decoded block by block through intra/inter predictions, DCT, and CA VLC. Deblocking filter is an efficient tool to reduce the blocking artifacts between the blocks generated by block-based tools listed ahead. Despite of the fact that deblocking filter is one of the most important tools of H.264/AVC, it is difficult to implement into software to perform real-time decoding or encoding of high resolution of video sequence due to the large computational complexity. Therefore, it is required to have an adaptive hardware for deblocking filter.

H.264/AVC has an advantage of transferring good-quality video in a low bit-rate. This video standard is used widely nowadays for the video-transfer in wireless communication and is the standard for Digital Multimedia Broadcasting (DMB) in Korea [3]. It is important to realize low-power video decoder because mobile applications have a limited battery capacity.

Ever since the announcing of the new video coding standard H.264/AVC, many researches have conducted designing deblocking filter architecture. Most architecture have been able to operate in a frequency above 100MHz using current fabrication technology. Furthermore, recently the filtering performance has improved to deliver a full HDTV (1920 x 1080) real-time video sequence. However, no work has been reported in realm of applying low-power deblocking filter in mobile applications. In this paper, it is proposed that a low-power architecture for the H.264/AVC deblocking filter which supports not only high resolution video such as full HDTV but also supports low-power mobile applications.

The rest of this paper is organized as follows: Section II describes the memory organization of the deblocking filter architecture. Section III reveals the detail of the proposed architecture. The result of the synthesis and the comparison with other architectures are discussed in section IV. Finally, conclusions are drawn in section V.

II. MEMORY ARCHITECTURE AND ORGANIZATION

The performance of deblocking filter depends on the memory architecture and organization. Well-organized memory architecture enables the implementation of the deblocking filter in the pipelined architecture of H.264/AVC decoder system with an enhanced performance. Memory architecture also affects power consumption where internal or external memory access acts as a significant factor in determining the power consumption.

A. Reconstructed memory

Our H.264/AVC decoder system was carefully designed to implement 4x4 sub-block level pipelined architecture. It is difficult to establish a direct connection with the deblocking filter with 4x4 sub-block level pipelined architecture because the reconstruction unit of the deblocking filter is a 16x16 macroblock. The reconstructed memory (Fig. 1) acts as a kind of bridge between 4x4 level and macroblock level pipeline. A 96x32 bit SRAM contains reconstructed pixel data of one macroblock, which are the results of inter/intra prediction are added with residual data from IDCT(Inverse DCT). Reconstructed memory consists of two single-port 96x32 bit SRAMs; one is for reading by deblocking filter and the other is for writing by residual adder. This double buffering enables the entire decoder architecture to simultaneously operate the 4x4 sub-block level pipeline while the deblocking filter running under the 16x16 macroblock level pipeline.
B. Adjacent memory

Deblocking filter not only requires the reconstructed pixels from the reconstructed memory but also requires of ready-filtered pixels when filtering the boundaries of macroblock. Fig. 1 represents these ready-filtered pixels, which is located adjacent to the current macroblock. As they are filtered pixels, the adjacent pixels should be delivered from the frame buffer through the data bus. However, the access to external memory decreases the performance of the deblocking filter and consumes a considerable amount of power. Employing a suitable size of SRAM could be the solution to this problem as SRAM can store the adjacent pixels [7] [8]. We used the term ‘adjacent memory’ to specify this particular SRAM for the rest of this paper.

The frame width, ‘W’, determines the size of the adjacent memory. The derivation of the equation used in Fig. 1 \((2W+32)\times32\) is as follows: ‘2W’ is used to cover the luminance and chrominance. The multiplying coefficient ‘32’ comes from the total number of bits which are divided into 4 pixels (8 bits each) vertically. The add coefficient ‘32’ is for the memory pixels stored in the left side adjacent memory. For example, to decode a CIF\((352\times288)\) size video sequence, a decoder requires a \((2\times352\times32)\times32\) bit SRAM.

III. PROPOSED ARCHITECTURE

The objective of this paper is to achieve the low-power operation of the deblocking filter architecture by followings: 1)enhancing the speed of filtering per macroblock, 2)reducing resources which consume large amount of power and 3)reducing the switching of register.

A. Hybrid Filtering Sequence

Filtering sequence defined in the standard of H.264/AVC [2] is not appropriate for recycling data because all horizontal edges should be filtered only after all vertical edges are filtered. It is known that many researches have excelled the architecture of recycling data by using novel filtering sequence. We adopted a hybrid filtering sequence which is introduced as ‘2-D processing order’ in [6]. Fig. 2 shows the filtering sequence of hybrid filtering. To begin with, vertical edge 1 is horizontally filtered before the filtering of vertical edge 2 occurs. Horizontal filtering and vertical filtering are then repeated alternatively for edge 3, 4, 5, 6 and 7. Horizontal edge 8 is filtered vertically by following the alternating sequence. This filtering sequence satisfies the required filtering condition defined in the standard of H.264/AVC, which states that the order of filtering 4x4 sub-block should follow the edge sequence of vertical-left, vertical-right, horizontal-upper and horizontal-bottom as well as enabling the recycle of data.

The architecture described in [6], with several internal SRAMs, requires a frequent access to the SRAMs causing a large power consumption. Another disadvantage of this architecture is that it requires a large chip area for deblocking filter. In order to solve this problem, a new architecture is proposed which reduces excessive power consumption while maintaining the same filtering sequence.

B. Deblocking Filter Architecture

The implemented hardware architecture for the deblocking filter is shown in Fig. 3. In Fig. 3, the bold line indicates deblocking filter and the memories outside of the bold line indicates SRAMs and frame buffer. Frame buffer becomes an external memory connected via the 32bit data bus. The proposed architecture consists of a 1-D FIR filter (Edge Filter), two 4x4 transpose arrays (Pbuf, Qbuf) to transpose the data for filtering and one register block (RegBlock) which contains four 4x4 buffers for a temporal store of the filtered data.

Fig. 4 is a detailed architecture of RegBlock, Pbuf and Qbuf. RegBlock is a temporal buffer for pixels which require one more filtering. One of the register blocks of RegBlock,
RB0-3, is stimulated to transfer data at one edge while the other register blocks keep their data. This means that no data transfer occurs in these register blocks. In other words, register transfer switching occurs at only one register block, giving an opportunity to reduce power consumption by clock gating scheme. Pbuf and Qbuf are 4x4 sub-block transpose matrices. These matrices transpose row-aligned pixels to column-aligned pixels or vice versa.

Fig. 5 shows the hybrid filtering sequence described earlier, where the macroblock is partitioned to reduce the number of registers and the number of access to SRAM(Adjacent memory). Each stage of partitioned macroblock consists of 8 edges, and the edges are filtered using four filtering modes - initial horizontal filtering (IHF), normal horizontal filtering (NHF), normal vertical filtering (NVF), and final vertical filtering (FVF).

In IHF mode to start with, left side of the edge block L0 comes from the adjacent memory and the right side of edge block B0 comes from the reconstructed memory. In one clock cycle, each of the four pixels of L0 and B0 are transferred as well as filtered by ‘Edge Filter’. Pixel transposing is not a necessary requirement to start IHF mode because L0 in adjacent memory is already aligned as a row of pixels as shown in Fig. 2. After completing the IHF mode, the filtered L0 and B0 are expressed as L01 and B01 respectively, which is the superscript 1 implies that the indicated block has been filtered once. Since L0 had undergone filtering for three times, it left only with one more filtering. As L01 is one step advanced filtered form than L0, it is transferred to frame buffer. Similarly, B01 gets stored in Pbuf and waits for the next filtering.

In NHF mode, B1 comes from the reconstructed memory and B01 comes from Pbuf. After completing NHF mode, B02 is stored into Qbuf, where B11 is stored in Pbuf. In NVF mode, B02 comes from Qbuf with a transposed form, and U0 comes from the adjacent memory. As shown in Fig. 2, U0 in the adjacent memory is already aligned in column, and hence NVF mode does not require any redundant clock cycle. After the NVF mode, U01 is transferred to frame buffer and B03 is stored into RegBlock. The NHF and NVF mode are then alternate twice. Lastly in FVF mode, U3 comes from the adjacent memory and B31 comes from Pbuf with a transposed form.

The filtering processes described above are applied to other stages in the same way. Filtered blocks \{B33, B73, B113, B123, B133, B143, B153\} are stored to the adjacent memory at the edges \{17, 25, 26, 27, 29, 31, 32\}. Block B123, B133, B143 and B153 are stored into the adjacent memory directly while B33, B73 and B113 require four clock cycles to be transposed from column-aligned pixels into row-aligned pixels, so that B33 and B73 can be stored at the edges \{17, 25, 32\} respectively. At edge 32, the filtered block B112 and B153 is required to be stored into the adjacent memory at the same time. This is because we need the additional four clock cycles to store the block B153 into the adjacent memory.

Chrominance filtering is almost the same as luminance filtering. We need 192 (i.e. 4x48) clock cycles to filter all of the horizontal and vertical edges in a macroblock, 12 clock cycles for the edges \{32, 40, 48\} for the data congestion and 4 extra clock cycles for transition of SRAM to read and write. Thus, there are a total of 208 clock cycles to filter a macroblock(Fig. 6). The access to the reconstructed memory, adjacent memory and frame buffer occurs 96, 64, and 96 times respectively in the process of one macroblock. When comparing to the other architectures referred in [4]- [9], our architecture offers minimum number of memory access. Considering that the SRAM access requires more power consumption than that of register switching, the proposed architecture reduces the power consumption.

IV. IMPLEMENTATION RESULTS

In order to test the performance of our architecture, we described the architecture into synthesizable verilog HDL and synthesized it using Synopsys Design Compiler with Samsung 0.18 µm cell library. The clock frequency is set to 125MHz and the result of synthesis shows that our architecture takes...
only 18.34K gate count. Synthesized design is verified with reference software [10].

Table I shows the comparison of our architecture with those in of [4]- [9]. The proposed architecture takes only 208 clock cycles per a macroblock while [4], [5] and [6] takes more than twice the clock cycles compared to the proposed architecture. This architecture shows a satisfactory performance, as 192 clock cycles is considered to be the optimal performance for deblocking filter with one 1-D FIR filter [8].

The proposed deblocking filter aims to achieve low-power consumption. In order to test for the power consumption, we extracted switching information of the synthesized architecture from simulations with several test sequences. We then simulated the power consumption using Synopsys Prime Power with the data from the switching information gathered previously. Fig. 7 shows our architecture drastically reduces power consumption using clock gating scheme. In ‘active’ mode, the power consumption of the clock-gated architecture is reduced by about 35%. In ‘sleep’ mode, the power consumption of the clock-gated architecture is further reduced by up to 92%. The drastic power reduction is the result of the decreased number of access to SRAM, the optimized register block architecture and the filtering process.

Fig. 8 clearly shows that the proposed architecture consumes the lowest power consumption. Due to the fact that the reference architectures [4]- [9] have different clock cycles to perform the deblocking filtering, the energy consumption of filtering a macroblock is chosen for comparison. The proposed architecture is simulated with synthesized architecture, while other references are estimated based on the register switching and SRAM access using Samsung 0.18µm standard cell library. The power consumption of 1-D FIR filter and the control unit are ignored because the reference architecture does not offer detailed architecture and switching information. Fig. 8 shows that the proposed architecture consumes the smallest energy per one macroblock.

V. CONCLUSION

We have proposed a low-power deblocking filter architecture for H.264/AVC. The design is implemented in synthesizable verilog HDL and the synthesized results take only 18.34K gate at 125MHz clock frequency with Samsung 0.18 µm standard cell library. The proposed architecture has shown the lowest power consumption without any degradation of filtering performance compared to the state-of-the-art architecture of deblocking filter.

REFERENCES


<table>
<thead>
<tr>
<th>Architecture</th>
<th>Cycle (MB)</th>
<th># of 4 x 4 (Arrays)</th>
<th>Clock (MHz)</th>
<th>Technology (µm)</th>
<th>Gate Count (K)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>208</td>
<td>6</td>
<td>125</td>
<td>0.18</td>
<td>18.34</td>
</tr>
<tr>
<td>[4]</td>
<td>614</td>
<td>4</td>
<td>100</td>
<td>0.25</td>
<td>20.66</td>
</tr>
<tr>
<td>[5]</td>
<td>50-342</td>
<td>2</td>
<td>100</td>
<td>0.18</td>
<td>11.8</td>
</tr>
<tr>
<td>[6]</td>
<td>446</td>
<td>8</td>
<td>100</td>
<td>0.25</td>
<td>24.0</td>
</tr>
<tr>
<td>[7]</td>
<td>250</td>
<td>4</td>
<td>100</td>
<td>0.18</td>
<td>19.64</td>
</tr>
<tr>
<td>[8]</td>
<td>192</td>
<td>2</td>
<td>200</td>
<td>0.13</td>
<td>7.5</td>
</tr>
<tr>
<td>[9]</td>
<td>214</td>
<td>2</td>
<td>100</td>
<td>0.18</td>
<td>20.9</td>
</tr>
</tbody>
</table>