REAL-TIME DEBLOCKING FILTER FOR MPEG-4 SYSTEMS

Hung-Chi Fang, Tu-Chih Wang, and Liang-Gee Chen

DSP/IC Design Lab
Graduate Institute of Electronic Engineering, National Taiwan University
NO. 1, Sec. 4, Roosevelt Road, Taipei 106, Taiwan
Phone: +886-23635251-332, Fax: +886-23681679
Email: {honchi, eric, lgchen}@video.ee.ntu.edu.tw

ABSTRACT

This paper presents a deblocking filter architecture in MPEG-4 standard. This architecture performs 1-D nonlinear filter across block boundary to efficiently suppress blocking artifacts. Two shift register banks are used in the design. The use of them greatly reduces the control complexity of data flow. Due to efficient scheduling, the size of storage and the total clock cycles are minimized. The maximum processing rate of this architecture mapped in 0.35 μm technology is 30 M pixels/sec, which can support NTSC resolution at 30 frames per second. The power consumption of the design is 46.6 mW while operating frequency is 81Mhz.

1. INTRODUCTION

In the DCT-based video coding standards, such as H.261 [1], H.263 [2], MPEG-2 [3] and MPEG-4 [4], annoying blocking artifacts rise when the compression ratio is high. This effect is caused by the non-overlapping nature of the block based DCT coding and the quantization of DCT coefficients. There are two categories of techniques to deal with blocking artifacts, one is to reduce its occurrence and the other is to remove it.

To reduce the blocking artifacts, the overlapped block motion compensation (OBMC) [5] is proposed and adopted into H.263 standard. Although OBMC can reduce blocking artifacts, encoder and decoder must support this mode simultaneously. And the computation requirement of encoder is much higher because OBMC motion estimation is hard to implement. On the other hand, many algorithms have been proposed to remove the blocking artifacts only in the decoder. The simplest one is using a low-pass filter to remove the artifacts by filtering across the block boundary. The drawback of the low-pass filter is that it introduces blurring effects on the real edges.

Iterative methods, maximum a *posteriori* (MAP) [6] and projection onto convex sets (POCS) [7], have better performance comparing to the simple low-pass filter algorithm. But the computational complexities of these two methods are very high and are not suitable for real-time application or devices with low computation power.

Kim et al. have proposed a nonlinear, adaptive filter to remove the blocking artifacts [8]. This algorithm performs well in DCT coding scheme while keeps computation complexity moderate. The proposed architecture in this paper is based on this algorithm, which has been adopted into MPEG-4 standard.

The remainder of this paper is organized as follows: In section 2, the algorithm of the deblocking filter is explained. The block diagram of the proposed architecture and the functionality of each module are illustrated in Section 3. The scheduling is shown in Section 4. Section 5 is the summary of the implementation results. Finally, concluding remarks are presented in Section 6.

2. DEBLOCKING FILTER ALGORITHM

The filter performs 1D filtering across the 8×8 block boundary. The filtering operation is first performed on horizontal edge and then vertical edge. The detailed algorithm of the deblocking filter is shown in Fig. 1. There are two operation modes in the algorithm. One is called "smooth mode" and the other is called "default mode." By examining the pixels across adjacent block boundaries, proper mode is chosen. If the gray values of the pixels are quite different with each other, the filter enters the default mode; otherwise it operates in the smooth mode.

Before switching into the smooth mode, we must detect whether blocking artifacts exist on the block boundary. If all the gray scales of pixels across the block boundary are within a threshold value, we can conclude that there is no blocking artifact on the boundary. And no filter is needed. The threshold is depended on quantization parameter (QP), which deter-

0-7803-7690-0/02/$17.00 ©2002 IEEE

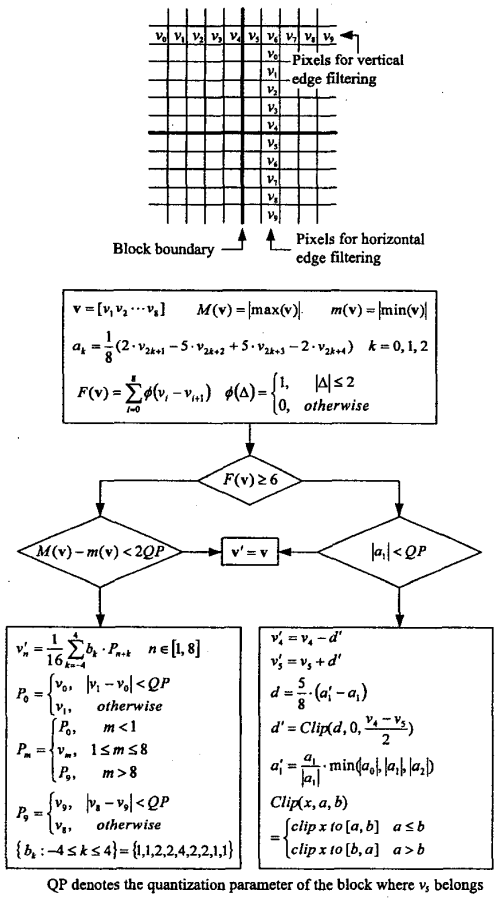

Figure 1: Flowchart of the deblocking filter algorithm.

mines coding distortions. The threshold value is set to max($2QP_l$, $2QP_r$), where QP_l is quantization parameter of the 8×8 block on the left side of the filtering boundary and QP_r represents it on the right side. When blocking artifacts are detected, a nine-tap low-pass filter is used to smooth the pixels across the boundary. When the pixel referenced by the nine-tap filter is outside the pixel scope (v_1 to v_8 in Fig. 1), padding operations are required.

The blocking artifacts detecting process in the default mode is not the same as in the smooth mode. When the magnitude of a_1, shown in Fig. 1, exceeds the value of QP, the default mode filter is performed.

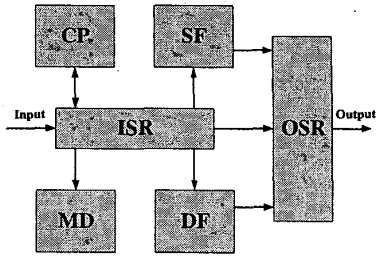

Figure 2: Block diagram of the proposed architecture.

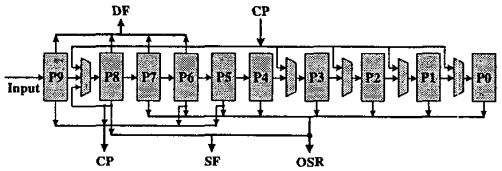

Figure 3: Interconnections between the ISR module and other modules.

3. SYSTEM ARCHITECTURE

The block diagram of the architecture is shown in Fig. 2. The input pixels are shifted into the ISR pixel-by-pixel. The MD module determines which mode should be used for the filtering operation. The DF module performs default mode function in Fig. 1, while SF module performs smooth mode. The CP generates padding pixels for filter modules. Filtered data is fed into OSR and sequentially shifted out. The filter can be integrated into encoder system after motion compensation unit. The output of the filter can be written into memory or display directly.

3.1. ISR MODULE

The function of ISR module is providing storage of input pixels for other modules. Fig. 3 shows the interconnections between the ISR and the other modules. Ten 8 bits width registers are used in this module. The data is read from the memory into "Input" port in Fig. 3 and sequentially propagates from $P9$ to $P0$. There are five multiplexers at the input of $P8$, $P3$, $P2$, $P1$ and $P0$. Padded data can be replaced in the correct location by these multiplexers. The schedule of the filter operations is kept tight so that the requirement of pixel storages can be reduced.

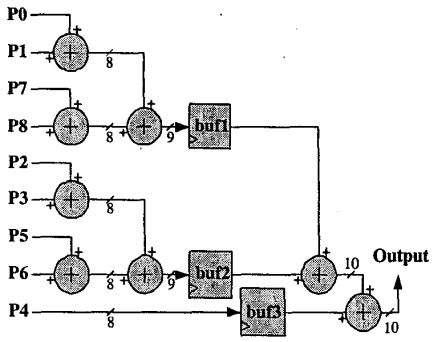

Figure 4: Circuit of the SF module.

3.2. MD MODULE

The function of MD is mode decision. It determines which filter should be used. The difference between two adjacent pixels is calculated and compared with a threshold. If the difference is larger than the threshold, the decision counter increases one. The decision is made when the last pixel is loaded.

3.3. DF MODULE

The DF module realizes direct mode filter. In this module, a two-stage pipeline architecture is used in order to reduce the critical path. The critical path is composed of a nine-bit and a ten-bit adder. Eight clock cycles is occupied for this module to complete the filtering. All the multiplications are replaced by shift-and-add since all the coefficients are simple integers.

3.4. SF MODULE

The smooth mode filter is mapped into the SF module. In this mode, a nine-tap low-pass filter is used to remove the blocking artifacts. The filter is realized by hardwired adders, as shown in Fig. 4. There is also a two-stage pipeline to cut the critical path. The critical path in this module becomes the cascading of two ten-bit adders that locates at final stage. Eight clock cycles is needed if this mode is active since eight output pixels need to be filtered and the throughput of this module is one.

3.5. CP MODULE

This module calculates padding pixels for the ISR module. The output of this module is used in place of the end pixel, i.e. v_0 or v_9. When v_0 is padded, the padding pixel will fill into P0-P3 simultaneously. On the other hand, when padding v_9, the padding pixel only replaces P8 register. Two clock cycles are needed for the CP module to complete a padding process.

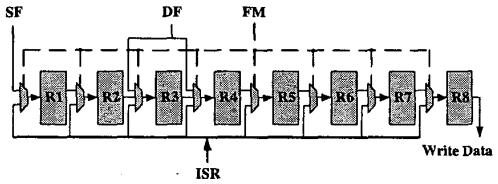

Figure 5: Interconnections between the OSR module and other modules.

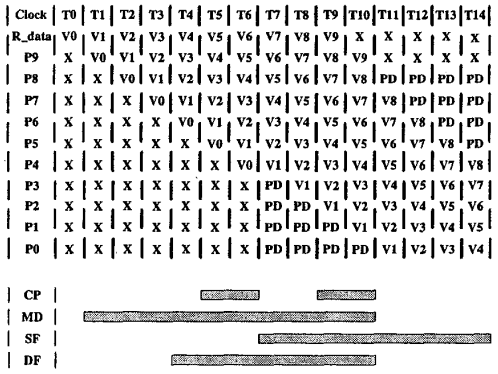

Figure 6: Data flow in ISR module and the operating period of every module.

3.6. OSR MODULE

As shown in Fig. 5, the inputs of the OSR are different according to the filtering mode. If the filtering mode is the smooth mode, output pixels are fed from R1 sequentially. In the default mode, the two filtered boundary pixels are input from R3 and R4 at the same clock. The other six pixels, which are left unchanged, come from the ISR. If there is no filtering operation, input pixels from the ISR fill into R0 to R8 concurrently. In all the three cases, the output is shift out via register R8 to overwrite original data in the memory. This design simplifies writing scheme of frame memory.

4. SCHEDULING

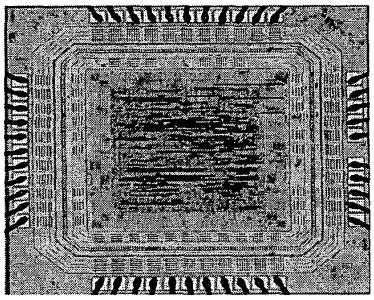

Figure 7: Die photo of the proposed deblocking filter.

The data flow in ISR is shown in Fig. 6. The symbol X implies "don't care." And PD denotes "padded," which is the result of the CP module. R_data is the input pixel read from memory.

Input pixels, v_0-v_9, are read from memory at T0-T9. The MD module operates from T1 to T10 and the decision is outputted at T11.

The DF acts at T4 and takes two clock cycles to complete a four point DCT. The second and third DCT coefficients are calculated at T6 and T8. After the third DCT coefficient is calculated, an extra clock cycle is occupied for computing the filtered values of v_4 and v_5. Finally, the output of the DF is ready at T11.

The CP module is triggered at T5 and T9 to generate padding pixels at T7 and T11, since there is a two clock cycle latency in it. The SF module starts at T7 following the first padding process. Eight clock cycles are required in this module.

Because this filter is adaptive, the total clock cycles needed are determined by filter mode used. In the default mode and no filtering mode, the total cycles are 19. In the case of smooth mode, 23 cycles are needed.

By carefully scheduling, there is no bubble in the pipeline and the total clock cycles are minimized.

5. IMPLEMENTATION RESULTS

Our chip is synthesized using 0.35 μm CMOS cell library. The die photo of the chip is shown in Fig. 7 and the chip specifications are listed in Table 1. The maximum clock rate is 100 MHz while the target working frequency is 81 MHz. The total gate count is 5213 and the transistor count is 29841. The die size is 1.778 $mm \times 1.778\ mm$. The power consumption is 46.6 mW at 83 MHz with supply voltage at 3.3 V.

6. CONCLUSIONS

Table 1: Specification of the chip.

Technology	TSMC 0.35 μm CMOS 1P4M
Package	68 PGA
Die size	1.778×1.778 mm^2
Logic gate count	5213
Transistor count	29841
Max. clock rate	100 MHz
Power consumption	46.6 mW @ 3.3 V, 83 MHz
Pad number	44
Input pad	24
Output pad	9

In this paper, the architecture of the MPEG-4 recommended deblocking filter is proposed. The use of two sets of shift registers makes the control of the data flow much easier and simpler. The chip consumes 46.6 mW at 83Mhz with the capability of NTSC 30fps real-time deblocking filter operation.

REFERENCES

[1] ITU-T, *Draft ITU-T Recommendation H.261*, Video codec for audio-visual services at p×64 kbits/s, 1993.

[2] ITU-T, *Draft ITU-T Recommendation H.263*, Video coding for low bitrate communication, 1997.

[3] ISO/IEC IS 13818(MPEG-2), *General Coding of Moving Pictures and Associated Audio Information*, 1994.

[4] ISO/IEC DIS 14496(MPEG-4), *Very-low Bitrate Audio-visual Coding*, Oct. 1998.

[5] M.l T. Orchard and G. J. Sullivan, "Overlapped Block Motion Compensation: An Estimation-Theoretic Approach," *IEEE Trans. Image Processing*, pp. 693–699, Sept. 1994.

[6] R. Stevenson, "Reduction of Coding Artifacts in Transform Image Coding," in *Proc. ICASSP-95*, Mar. 1995, pp. 2363–2366.

[7] R. Rosenholtz and A. Zakhor, "Iterative Procedures for Reduction of Blocking Effects in Transform Image Coding," *IEEE Trans. Circuits Syst. Video Technol.*, pp. 91–95, Mar. 1992.

[8] S. D. Kim, J. Yi, and J. B. Ra, "A Deblocking Filter with Two Separate Modes in Block-Based Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, pp. 156–161, Feb. 1999.