5th International Conference on VLSI and CAD
October 13 - 15, 1997, Seoul, Korea

A MEMORY-BASED PIPELINED ARCHITECTURE FOR BLOCKING EFFECT REMOVAL IN HDTV

Jae-Wook Lee, Myung-Hoon Yang, Sungho Kang and Yoonsik Choe
Dept. of Electrical Engineering, Yonsei University
Shinchon-Dong 134, Seodaemoon-Gu, Seoul, Korea
E-mail: shkang@bubble.yonsei.ac.kr, Phone: 82-2-361-2775, Fax: 82-2-313-8053

Abstract

This paper presents an efficient memory-based architecture for blocking effect removal in HDTV. To reduce the size of a memory, a memory is partitioned into many memory banks. This makes it possible to access the memory concurrently. Also, to improve the operation speed, a pipelined parallel architecture and a memory scheduling technique are adopted. Since multiplications and divisions are time-critical, these operations are replaced with shiftings. Therefore this architecture is very fast and uses small size memory banks, and this makes it possible to realize a real-time signal processor.

1 Introduction

In digital HDTV, DCT(Discrete Cosine Transform) is used to reduce the image data before broadcasting image signals, and then the received image signals are restored by IDCT(Inverse Discrete Cosine Transform). But owing to the characteristics of these process, we can see the discontinuity between 8×8 image blocks in the restored image. These phenomenon is called a blocking effect.

This blocking effect can be removed by low-pass filtering, but low-pass filtering blurs the edges of the image. Therefore using an improved algorithm[1] the direction of an edge is classified. According to the direction of the edge only the boundary pixels of the image blocks are filtered with various filter coefficients to avoid the edge-blurrings.

In this paper, a memory-based pipelined parallel architecture[2] is used to implement a fast signal processing. With this architecture, blocking effect removal was simulated with C language, and an ASIC(Application Specific Integrated Circuit) chip was realized with VHDL(VHSIC Hardware Description Language)[4].

2 Algorithm

1. Let two counters K and L be set to zero. And let $x_{i,j}$ denote the (i,j)th pixel in an 8×8 image block, and let x_{avg} denote the average value of two values.

2. Let τ denote the preselected threshold. For all horizontally adjacent pixels,

$$\Delta x_{i,j} = x_{i+1,j} - x_{i,j},\ x_{avg} = \frac{x_{i+1,j} + x_{i,j}}{2}$$

$$where, i = 1, \cdots, 7 \ \ and \ \ j = 1, \cdots, 8$$

If $\frac{\Delta x_{i,j}}{x_{avg}} > \tau$ then increase the K counter by 1.
If $\frac{\Delta x_{i,j}}{x_{avg}} < -\tau$ then decrease the K counter by 1.

3. For all vertically adjacent pixels,

$$\Delta x_{i,j} = x_{i,j+1} - x_{i,j},\ x_{avg} = \frac{x_{i,j+1} + x_{i,j}}{2}$$

$$where, i = 1, \cdots, 8 \ \ and \ \ j = 1, \cdots, 7$$

If $\frac{\Delta x_{i,j}}{x_{avg}} > \tau$ then increase the L counter by 1.

If $\frac{\Delta x_{i,j}}{x_{avg}} < -\tau$ then decrease the L counter by 1.

4. Let m denote the minimum length of an edge segment to be a meaningful edge. Using the counted two values of K and L, classify the direction of an edge.

   (a) Monotone : if $|K| < m \ \ and \ \ |L| < m$
   (b) 0° edge : if $|K| < m \ \ and \ \ |L| > m$
   (c) 45° edge : if $|K| > m, \ \ |L| > m,$ $and \ \ sgn(K) = sgn(L)$
   (d) 90° edge : if $|K| > m \ \ and \ \ |L| < m$
   (e) 135° edge : if $|K| > m, \ \ |L| > m,$ $and \ \ sgn(K) = -sgn(L)$

By experiments the values of 'm' and 'τ' were determined to $m = 6 \ \ and \ \ \tau = 0.2$.

3 Memory Scheduling and Partitioning

In this architecture, six memory modules are used, and each memory module is composed of ten memory banks. Nine memory banks are used to store input image signals, and a bit large size memory bank is used to store filtered image signals.

Each module is used for processing 8 lines of pixels in HDTV, because we decide the edge direction of an image block based on an 8×8 image block.

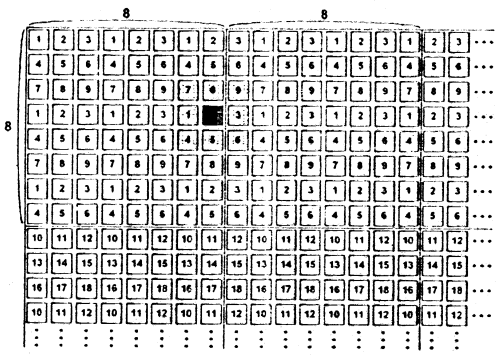

Figure 1: The memory bank assignments in a HDTV image

In Figure 1, a small block means a pixel in a HDTV screen, and the number in it represents the memory bank number for storing that image signal. And a shaded 3×3 block represents memory banks which are used for a 2D-filtering to produce a filtered result for a heavily shaded pixel. Although the filter mask moves to another position, nine image signals can be accessed by using nine memory banks simultaneously. This can solve the problem of a memory access bottleneck.

	Step1	Step2	Step3	Step4	Step5	Step6	Step7	Step8	Step9	Step10	...
Module1	Op 1	Op 2	Op 3	wait	Op 4	Op 5	Op 1	Op 2	Op 3	Op 4	...
Module2		Op 1	Op 2	Op 3		Op 4	Op 5	Op 1	Op 2	Op 3	...
Module3			Op 1	Op 2	Op 3		Op 4	Op 5	Op 1	Op 2	...
Module4				Op 1	Op 2	Op 3		Op 4	Op 5	Op 1	...
Module5					Op 1	Op 2	Op 3		Op 4	Op 5	...
Module6						Op 1	Op 2	Op 3		Op 4	...

where, Op1 : writes input data to memory banks
Op2 : decides the edge direction of a block
Op3 : filters the horizontal border of a block
Op4 : filters the vertical border of a block
Op5 : reads output data from memory banks

Figure 2: A memory scheduling

Figure 2 shows a memory scheduling[3]. The states of memory modules are shown in each time steps.

In this architecture, 54 1920-byte memory banks and 6 6720-byte memory banks are required. That is, the amount of $54 \times 1920 + 6 \times 6720 \simeq 141(Kbytes)$ total memory is used.

4 Architecture

Figure 3 shows the architecture for removing the blocking effect. First,input image signals are stored into memory banks by a memory mapper for signal input. And then computations are performed to produce counter control signals by an edge detector. The numbers of K and L values are counted, and the edge directions of an 8×8 image block is determined by a direction detector. The right and left boundary pixels of an 8×8 image block are filtered by a horizontal filter, and the upper and lower boundary pixels of an 8×8 image block are filtered by a vertical filter. Finally, the filtered images are sent to an output port by a memory mapper for a signal output. In the following, sub-blocks are explained in detail.

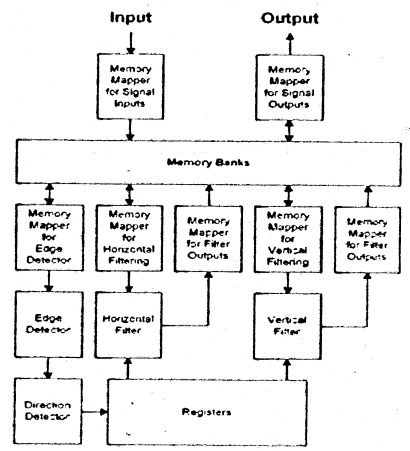

Figure 3: The architecture for blocking effect removal

An edge detector is used to perform the following kind of computations.

$$\frac{\Delta x_{i,j}}{x_{avg}} = \frac{x_{i+1,j} - x_{i,j}}{x_{i+1,j} + x_{i,j}} \times 2 > \tau$$

To reduce the calculation time, the equation is modified like the following equation.

$$2 \times (x_{i+1,j} - x_{i,j}) > \tau \times (x_{i+1,j} + x_{i,j})$$

To optimize processing time, $0.25(=\frac{1}{4})$ is selected as τ instead of 0.2. The final equation to calculate is presented below.

$$2^3 \times (x_{i+1,j} - x_{i,j}) > (x_{i+1,j} + x_{i,j})$$

The multiplication of 2^3 means 3-bit left shifts, so instead of multiplication and division, shifts can be used.

In an edge detector, there are two identical blocks. One is for producing L value control signals, and the other is for producing K value control signals.

A direction detector is used for counting the numbers of K and L values, and then for comparing the result based on the algorithm step 4 which is mentioned in section 2. After deciding the edge direction of an image block, the results are encoded.

In this proposed architecture, filters are the most time-critical blocks. In a 2-D 3×3 low-pass filtering, 9 image pixels are used at the same time. Because a blocking effect takes place at the boundary of each image block, only 28 boundary pixels of each 8×8 image block are filtered.

To realize the real-time processing, calculation processes must be simplified. As shown in Figure 4, floating-point filter coefficients are replaced by integer coefficients. The integer coefficients are made by multiplying floating-point filter coefficients by $1024(=2^{10})$. After the calculations, the desired results can be taken by 10-bit right shiftings. To replace multiplying processes with shifting processes, produced integer coefficients are replaced by the closest $2^n(n = 1, 2, 3, \cdots)$ values.

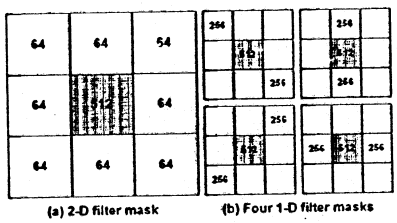

Figure 4: Filter Masks

In filters, according to the edge direction, an amount-selector determines the amount of bits for shiftings. Input image signals are shifted by shifters with the help of amount-selector outputs. Since the shifting-amounts are fixed to several values, the shifting-operations can be realized by extracting the required bits from the intermediate signal values and then wiring them into the expected position of result values. These processes are used instead of using floating-point multipliers, which greatly reduces the calculation time.

Shifted nine input signals are added by a wallace-tree[6]. This reduces the addition steps, and realizes fast addition of many values.

After reduction, two values are added by a carry-select adder which is fast for an addition of two values. And the result is shifted by 10 bits to the right direction to compensate the filter coefficients which have been shifted by 10 bits to the left direction.

In mappers, all the I/O ports are driven by many tri-state buffers, and all the tri-state buffers are controlled by a decoder block. The decoder block enables the corresponding tri-state buffers for memory bank enable ports, for address ports, and for data ports. According to the CLK signal, the counters decide the addresses and the memory banks for signal processings in each time-step.

5 Conclusions

In this proposed architecture for blocking effect removal, to reduce the memory size and to overcome the memory access bottleneck, the efficient ways of memory scheduling and memory partitioning are devised. A single memory is partitioned into 60 small memory banks, and all memory banks can be used at the same time to reduce the memory access time.

In the calculation process, all the floating-point coefficients are replaced by integer coefficients using shifting operations to simplify the architecture and to reduce the processing time. And, all the multiplications and divisions are realized by using modified coefficients and shifters.

So this architecture is quite fast enough to realize the real-time processing, and uses small size memory. As a result, this architecture is efficient for realizing a real-time signal processor.

References

[1] B. Ramamurthi and A. Gersho, "Nonlinear Space-Variant Postprocessing of Block Coded Images," in *IEEE Trans. on Acoustics, Speech, and Signal Processing*, October 1986, pp. 1258–1267.

[2] T. Ishiguro, *Signal Processing of HDTV*. North-Holland: Elsevier Science Publishers, 1988.

[3] J. L. Hennessy and D. A. Patterson, *Computer Architecture - A Quantitative Approach*. Morgan Kaufmann Publishers, 1996.

[4] Z. Navabi, *VHDL : Analysis and Modeling of Digital Systems*. USA: McGRAW-HILL, 1993.

[5] C. L. Seitz, "Concurrent VLSI Architectures," in *IEEE Trans. on Computers*, 1984, pp. 16–34.

[6] C. S. Wallace, "A Suggestion for a Fast Multiplier," in *IEEE Trans. on Electronic Computers*, February 1964, pp. 14–17.