Abstract

This paper presents an efficient architecture for blocking effect removal in HDTV. Since there is a lot of image signal for signal processing in digital HDTV, the memory size and the fast operation have been the main concerns of the DSP(Digital Signal Processing) architectures. 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

With the invention of television we could see the images and hear the sounds of all of the world at any time, but the first invention could display only monochrome images without colors. We longed for color images and finally invented the color television, but it was not a satisfactory one. We also wanted stereo sounds, and made a stereo sound television. Now, we want cinema quality images which are large, wide and clear. At last, HDTV(High Definition Television) has been invented, but there are many problems in realizing a piece of commercial goods which is not expensive but still has high quality.

Early HDTV was designed to get broadcasting signals via broadcasting-satellite channels which have wide frequency band-width. Because HDTV has quite a lot of image signal compared with normal TV, wide frequency band-width is necessary. But for commercial purposes, broadcasting via narrow frequency band-width is necessary. In digital HDTV, image signals are compressed by DCT(Discrete Cosine Transform) and then broadcasted via existing TV broadcasting channels. Broadcasted signals are caught and then restored by IDCT(Inverse Discrete Cosine Transform). In this process of compression and restoration, data losses are unavoidable, so there exist a few differences between restored images and original images. In DCT process, 8×8 size image blocks are used. If we compare a restored 8×8 image block with an original one, it is difficult to find the differences. However, if we gather many restored 8×8 image blocks and compose a large image, we can see the discontinuity between each block. These phenomena are called a blocking effect and a staircase effect which result in image-quality degradations.

This blocking effect can be removed by low-pass filtering, but low-pass filtering blurs the edges of the image. Therefore using Ramamurthi's 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[2,3,4] is presented for blocking effect removal. Because there is too much data to process in HDTV, a pipelined parallel architecture is necessary to implement a real-time architecture for blocking effect removal. To reduce the memory size and to realize an easy and fast access to a memory, memory-scheduling and many memory banks are used[5,6]. In the process of blocking effect removal, many multiplications and divisions are needed. However these operations are time-critical[7], so the multiplications and divisions are replaced with shifters to realize a real-time 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)[8,9,10].

This paper is organized as follows. In section 2, a blocking effect removal algorithm is explained. Sec-
tion 3 treats the memory scheduling and partitioning. In this section, the way to reduce memory size for signal processing and the assignment of each image pixel to memory banks for concurrent accesses to memories are discussed. In section 4, the global architecture for blocking effect removal is presented and the subblocks are explained in detail. In this section, the operations and the characteristics of all the blocks are presented.

2 Algorithm

In this paper, a blocking effect removal algorithm based on Ramamurthi’s algorithm[1] is used. This algorithm is used to determine the direction of an edge in an 8×8 size image block and then based on the determined direction of the edge the low-pass filter coefficients are determined not to blur the edge of each 8x8 image block. Because a blocking effect takes place at the boundary of each image block, only 28 boundary pixels of each 8x8 image block are filtered.

The suggested algorithm is as follows:

1. Let two counters \( K \) and \( L \) be set to zero. And let \( x_{i,j} \) denote the \((i,j)\)th pixel in an 8x8 image block, and let \( x_{avg} \) denote the average value of two values.

2. Let \( \tau \) denote the preselected threshold. For all horizontally adjacent pixels, perform the following computations.

\[
\Delta x_{i,j} = x_{i+1,j} - x_{i,j}
\]

\[
x_{avg} = \frac{x_{i+1,j} + x_{i,j}}{2}
\]

where, \( i = 1, \ldots, 7 \) and \( j = 1, \ldots, 8 \)

If the computation results satisfy the equation of \( \frac{\Delta x_{i,j}}{x_{avg}} > \tau \) then increase the \( K \) counter by 1.

If the computation results satisfy the equation of \( \frac{\Delta x_{i,j}}{x_{avg}} < -\tau \) then decrease the \( K \) counter by 1.

3. For all vertically adjacent pixels, perform the following computations.

\[
\Delta x_{i,j} = x_{i,j+1} - x_{i,j}
\]

\[
x_{avg} = \frac{x_{i,j+1} + x_{i,j}}{2}
\]

where, \( i = 1, \ldots, 8 \) and \( j = 1, \ldots, 7 \)

If the computation results satisfy the equation of \( \frac{\Delta x_{i,j}}{x_{avg}} > \tau \) then increase the \( L \) counter by 1.

If the computation results satisfy the equation of \( \frac{\Delta x_{i,j}}{x_{avg}} < -\tau \) 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. The classifications are shown below.

(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 \( \text{sgn}(K) = \text{sgn}(L) \)

(d) 90° edge : if \( |K| > m \) and \( |L| < m \)

(e) 135° edge : if \( |K| > m \), \( |L| > m \), and \( \text{sgn}(K) = -\text{sgn}(L) \)

By experiments the values of ‘\( m \)’ and ‘\( \tau \)’ were determined to \( m = 6 \) and \( \tau = 0.2 \). Using the results from above algorithm, filter coefficients are determined and then used for that image block. For a monotone block, a 2D low-pass filter which has filter coefficients like the followings is used.

\[
h(0,0) = 0.547
\]

\[
h(0,1) = h(0,-1) = h(1,0) = h(-1,0) = 0.227
\]

For a block which has a distinct edge-direction, a 1D low-pass filter which is parallel to the direction of the edge is used. The filter coefficients are shown below.

\[
h(0) = h(4) = 0.036,
\]

\[
h(1) = h(3) = 0.282,
\]

\[
h(2) = 0.363
\]

3 Memory Scheduling and Partitioning

In digital HDTV, an image screen is composed of 1920×1080 pixels. So 1920 × 1080 ≈ 2(Mbytes) memory is needed to store a screen when we consider only the Y signal among Y,U,V signals. And to store both input signals and filtered signals, at least 2 × 2 = 4(Mbytes) are needed in the end. Because there are many modules which require memories in HDTV, a 4 Mbytes memory is a very large size only for blocking effect removal. In DSP, since frequent memory accesses are necessary, the memory size and fast memory accesses are very important.

In this paper, the new memory scheduling and partitioning techniques are presented to solve these problems. To reduce the memory size and to make it possible to access a memory concurrently, a memory is partitioned into many memory banks.
Figure 1: The structure and partitioning of memory banks

Figure 1 shows a partitioning & structure of memory. 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.

Because nine pixels are needed simultaneously in filtering process, nine memory banks are used for storing input image signals in each module. And 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. So after storing 8 lines of image pixels, we can start image processing.

Figure 2 shows an assignment of memory banks to each pixel in a HDTV image. 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. As shown in Figure 2, 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. So nine pixel-signals are needed simultaneously. To prevent a bottleneck effect in memory accesses, 9 memory banks are used. 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.

Figure 3 shows a memory scheduling. The states of memory modules are shown in each time steps. After 6 time-steps, 6 memory modules are used at the same time. This speeds up the memory access and greatly improves the overall performance of signal processings. In a blocking effect removal process, 5 operations are needed. The first operation is to write input signals into memory banks, the second operation is to decide the edge direction of a block, the third operation is to filter the horizontal border of a block, the fourth operation is to filter the vertical border of a block, and the fifth operation is to read result data from memory banks.

Each bank stores 3 lines and each line is stored by 3 memory banks, and a line is composed of 1920 pixels. So each memory bank size must be 1920×3 = 1920(bytes). And a memory module can store 240 8×8 image blocks. In each block, boundary 28 pixels are used for filtering. So a 240×28 = 6720(bytes) memory bank for filtered results is needed. In brief, 54 1920-byte memory banks and 6 6720-byte memory banks are used in this architecture. That is, the amount of 54×1920+6×6720 ≈ 141(Kbytes) total memory is used. This amount is very small compared with 4(Mbytes) in a normal architecture.

Owing to the small amount of memory, memory banks can be placed in an ASIC chip. This greatly reduces the memory access time, and improves the overall performance of a DSP chip.
4 Architecture

As presented in the previous sections, to realize an architecture for blocking effect removal, a memory-based pipelined parallel architecture is required[11,12]. To reduce the memory size and to achieve a fast access to memory, a single memory is partitioned into many memory banks. But in the partitioned memory-bank architecture, the control of memory access is the most important and difficult thing. In this architecture, memory mappers control the memory access and memory scheduling.

![Diagram](image)

Figure 4: The architecture for blocking effect removal

As shown in Figure 4, seven memory mappers are used for blocking effect removal. All seven memory mappers operate at the same time, and they have counters for making memory addresses and memory banks selections. An edge detector is used to perform computation and to generate counter control signals. A counter is used to count the number of $K$ and $L$ values, and to determine the edge directions of $8 \times 8$ image blocks. Registers are used to store the determined edge directions of image blocks. And a horizontal filter is used for filtering right and left boundary pixels of an $8 \times 8$ image block when we consider horizontally arranged image blocks. A vertical filter is used for filtering upper and lower boundary pixels of an $8 \times 8$ image block when we consider vertically arranged image blocks.

The operation of this architecture is as follows:

1. Input image signals are stored into memory banks by a memory mapper for signal input.
2. Computations are performed to produce counter control signals by an edge detector.
3. The numbers of $K$ and $L$ values are counted, and the edge directions of an $8 \times 8$ image block is determined by a counter block.
4. The right and left boundary pixels of an $8 \times 8$ image block are filtered by a horizontal filter.
5. The upper and lower boundary pixels of an $8 \times 8$ image block are filtered by a vertical filter.
6. The filtered images are sent to an output port by a memory mapper for a signal output.

In the following sub-sections, sub-blocks are explained in detail.

4.1 Edge detector

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, $0.25 (= \frac{1}{2})$ is selected as $\tau$ instead of 0.2, which result in a little quality.
degradation of an image. However, it is acceptable, since the image degradation is so small that it can not be recognized. 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.

The architecture of an edge detector is shown in Figure 5. A subtractor, an adder, and a shifter are used for each input to perform equation calculation. 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.

### 4.2 Direction Detector

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. The architecture of a counter is shown in Figure 6, and the produced encodings are shown in Table 1.

#### 4.3 Filters

In this proposed architecture, filters are the most time-critical blocks. In a 2-D \(3 \times 3\) low-pass filtering, 9 image pixels are used at the same time. The low-pass filtering equation is as follows.

\[ P_{2,2}^{\text{rew}} = P_{1,1} \times M_{1,1} + P_{1,2} \times M_{1,2} + P_{1,3} \times M_{1,3} + P_{2,1} \times M_{2,1} + P_{2,2} \times M_{2,2} + P_{2,3} \times M_{2,3} \]

In this equation, \(P_{i,j}\) denotes the \((i,j)\)th image pixel in a selected \(3 \times 3\) image block, and \(M_{i,j}\) denotes the \((i,j)\)th filter coefficient in a filter mask. As shown in the above equation, 8 additions and 9 multiplications are needed. In addition, the multiplicands are floating-point numbers, so it takes more time than integer numbers in calculation. To realize the real-time processing, calculation processes must be simplified. As shown in Figure 7, 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 7(a) represents 2-D low-pass filter mask, and Figure 7(b) represents four 1-D low-pass filter masks. According to the edge direction, the kind of a 1-D low-pass filter is determined.

<table>
<thead>
<tr>
<th>Encodings</th>
<th>Edge direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>monotone</td>
</tr>
<tr>
<td>001</td>
<td>0° edge</td>
</tr>
<tr>
<td>010</td>
<td>45° edge</td>
</tr>
<tr>
<td>011</td>
<td>90° edge</td>
</tr>
<tr>
<td>100</td>
<td>135° edge</td>
</tr>
<tr>
<td>111</td>
<td>complicated edge</td>
</tr>
</tbody>
</table>

Table 1: The encodings of the edge directions

\[ +P_{3,1} \times M_{3,2} + P_{3,2} \times M_{3,2} + P_{3,3} \times M_{3,3} \]

Figure 8 shows the architecture of a filter. Nine image inputs are used for filtering. And 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. These processes are used instead of a floating-point multiplier, which greatly reduces the
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. Using this architecture, the memory size has been reduced from 4(Mbytes) to 141(Kbytes), and the memory access bottleneck problem has been solved.

In the calculation process, all the floating-point numbers are replaced by integer numbers using shifters 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.

### 4.4 Memory mappers

Figure 9 shows the architecture of a memory mapper. In an architecture for blocking effect removal, there are seven memory mappers. They are only different from each other in the direction of signals and the algorithms of counter operations. Basically, 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 HDTV image processings, there is quite a lot of image signal compared with normal TV. Due to the massive amount of image signals, the size of memory and the processing time have been the main concerns of the DSP architecture.

References


Biographies

**Jae-Wook Lee** is a M.S. student in the Department of Electrical Engineering at Yonsei University, Korea. He received a B.S. degree in Electrical Engineering from Yonsei University in 1996. His research interests include Computer Systems, Parallel Architecture, Image Processing, ASIC Design, Design for Testability, and BIST.

**Myung-Hoon Yang** is a M.S. student in the Department of Electrical Engineering at Yonsei University, Korea. He received a B.S. degree in Electrical Engineering from Yonsei University in 1996. His research interests include DSP Chip Design, VLSI CAD, Design for Testability.

**Sungho Kang** is an Assistant Professor in the Department of Electrical Engineering at Yonsei University, Korea. He received a B.S. degree from Seoul National University, Korea and the M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Texas in Austin, respectively. He was a senior staff engineer at the Semiconductor Systems Design Technology, Motorola Inc., a research scientist at Schlumberger Laboratory for Computer Science and a post doctoral fellow at the University of Texas at Austin. Dr. Kang's research interests include Testing and Design for Testability, VLSI CAD, and VLSI Design.

**Yoonsik Choe** received the B.S degree in electrical engineering from Yonsei University, Korea, in 1979, the M.S.E. degree in systems engineering from Case Western Reserve University, Cleveland, OH, in 1984, the M.S. degree in electrical engineering from Pennsylvania State University, University Park, in 1987, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, in 1990. He worked at Hyundai Industrial Electronics Research and Development Center as a member of the senior technical staff. Since 1993, he has been an assistant professor of Department of Electrical Engineering at Yonsei University, Korea. His current research interests include image analysis and processing, computer vision, pattern recognition, and HDTV. Dr. Choe is a member of Eta Kappa Nu.