AN AREA EFFICIENT DCT ARCHITECTURE FOR MPEG-2 VIDEO ENCODER

Kyeounsoo Kim and Jong-Seog Koh

Abstract — This paper presents an area efficient VLSI architecture of transform coding module for MPEG-2 video encoder. This module consists of 2-D DCT and 2-D IDCT, Q and IQ, and zigzag and alternate scan conversion circuits. Hardware cost and performance of this module are mainly affected by 2-D DCT and 2-D IDCT. In the proposed architecture, it would be shown that a single 1-D DCT/IDCT could take roles of 2-D DCT and 2-D IDCT. It is capable of reusing a single 1-D DCT/IDCT four times. It is based on the row-column decomposition technique. It can be achieved through precise timing schedules. Intuitively, three 1-D DCT/IDCT and a matrix transposition memory could be saved as compared to the conventional architectures, which usually use two one-dimensional transforms and transposition memory. Even though there are some extra circuits due to timing controls and processing sequence schedules, this architecture takes about 24% and 50% respectively less area than the architectures published in [2] and [3]. This design and implementation are applicable to MPEG-2 video encoder accepting NTSC and PAL image formats in which the number of clocks to be allocated during a macro block period is 1320 for 54 MHz operating clock. To reduce its processing time, the proposed architecture uses 3-bit serial distributed arithmetic method. As a result, this architecture can be characterized to maximize the utilization of the hardware resources, and can be utilized for the encoders having similar structure as the MPEG-2 video encoder. It also can be applied to the ASIC chips for multimedia services especially requiring low hardware complexity.

Index terms — transform coding, MPEG-2 video encoder, discrete cosine transform

I. INTRODUCTION

There are two paths for transform coding in MPEG-2 video encoder, as illustrated in Figure 1. 2-D DCT and 2-D IDCT inside the dotted line of the figure are designed utilizing only a single 1-D DCT/IDCT. For real time processing, it is assumed that this module should process 384 data within a macro block that consists of 1320 cycles (54 MHz) for NTSC and PAL image formats [9,10]. Since these operations should be performed sequentially as the former results are used for the later inputs, the processing time is not sufficient to make a single 1-D DCT/IDCT process 6 blocks of 384 data during a macro block period. It can be achieved through precise timing schedules and improving throughput of 1-D DCT/IDCT. If these procedures can be covered with 1-D DCT/IDCT within the specified period, large hardware size can be saved. It is possible to use 1-D DCT/IDCT four times repeatedly by the orthogonal and separable properties of DCT.

Until now, many efficient VLSI architectures for the area and processing speed points of views have been reported [1-6]. Among the existing architectures, many hardware design engineers have used distributed arithmetic and hardwired multiplier based architectures. Sun et al. proposed 16x16 DCT/IDCT using distributed arithmetic technique based on 1-bit at a time bit-serial processing in [4]. Uramoto et al. implemented 8x8 DCT/IDCT core processor operating at 100 MHz in [5]. They used 2-bit at a time bit-serial distributed arithmetic technique. These processors used two 1-D DCT/IDCT and a transposition memory. Medisetti et al. proposed a new idea to implement 100 MHz 2-D DCT/IDCT using one 1-D DCT/IDCT and a transposition memory[6]. They devised the hardwired-multiplier architecture for multiplying input pixel data by constant DCT coefficients. Distributed arithmetic technique takes advantages of being capable of expending 1-bit serial to two or more bits serial methods as shown in [7]. Distributed arithmetic technique can be easily adopted to improve throughput of 1-D DCT/IDCT. 2-D DCT/IDCT architecture in [6] is already well optimized in architectural level. Thus, in the proposed architecture the above advantages of distributed arithmetic and 2-D DCT/IDCT architecture using one 1-D DCT/IDCT is adopted for high throughput and low complexity.

The processors in [5] and [6] consumed 102 K and 67 K transistors, respectively. It means that the architecture using one 1-D DCT/IDCT is more efficient than that using two 1-D DCT/IDCT. In the proposed architecture, 101.3 K transistors are consumed. When the architectures in [5] and [6] are utilized for the MPEG-2 video encoder, they should be used twice for transform coding. That is, 204 K and 134 K transistors, and two transposition memories should be used. The proposed architecture consumes about 50% less resources than the architecture in [5], and 24% for the architecture in [6].
Figure 2. The proposed architecture of transform coding module

2) The data are read from the VLC buffer. The read addresses are also generated in SAG. After inverse quantization, the transformed data are multiplexed with input pixel data. The results are produced by the same procedures in 2-D DCT except for the difference between forward and inverse transforms.

The final results of the inverse transform are stored to the buffer in ADDER. There are also needed buffers in ADDER for interfacing between macro blocks. These kinds of buffers in VLC and ADDER are needed to construct MPEG-2 video encoder architecture. In the proposed architecture, the buffers are commonly used for intermediate data between two macro blocks as well as internal data within a macro block. Moreover, since the buffers in VLC can also be utilized for scan order conversions, it is of no use containing a 8x8 block data for scan conversions. As there is a macro block timing difference in the DPCM loop, input data interface can also be achieved by buffer interface. It means that our transform coding module should be controlled by the specific speed and timing schedule.

2.2 Architecture of the 1-D DCT/IDCT Unit

In the proposed 1-D DCT/IDCT architecture, the matrix-vector multiplication form of the DCT algorithm is referred. 2-D DCT and IDCT can be designed by row-column decomposition technique, which performs first 1-D DCT/IDCT transform, matrix transposition, and second 1-D DCT/IDCT transform sequentially. One-dimensional transform formulae of two-dimensional transform are divided into eight sets of matrix-vector multiplications. The hardware architecture of each matrix-vector multiplication can be implemented by four multiplications and their accumulations. In addition to them, eight adders/subtractors are needed in the input and output sides according to DCT/IDCT control. For 8x1 DCT, these operations are shown in (1) and (2).

In (1) and (2), the coefficients within the matrix-vector formulae are defined in (3), and they are real values.
\[
\begin{align*}
\vec{y}_6 &= \begin{bmatrix} a & a & a & a \end{bmatrix} \begin{bmatrix} x_0 + x_1 \\ x_1 + x_2 \\ x_2 + x_3 \\ x_3 + x_4 \end{bmatrix} \\
\vec{y}_3 &= \begin{bmatrix} b & d & e & g \\ d & -g & -b & -e \\ e & -b & g & d \\ g & -e & d & -b \end{bmatrix} \begin{bmatrix} x_0 - x_1 \\ x_1 - x_2 \\ x_2 - x_3 \\ x_3 - x_4 \end{bmatrix} \\
\vec{y}_9 &= \begin{bmatrix} \cos(\pi/2) \\ \cos(\pi/4) \\ \cos(\pi/8) \\ \cos(\pi/16) \\ \cos(\pi/18) \\ \cos(3\pi/16) \\ \cos(3\pi/18) \\ \cos(3\pi/16) \\ \cos(7\pi/16) \end{bmatrix}
\end{align*}
\]

For the 1-D IDCT, input data are divided into even and odd components, and they are multiplied by coefficients, and the results are accumulated, and finally each of the data are summed and subtracted as in (4) and (5). Since formulae, (1), (2), and (4), (5) are symmetric for even and odd components, they can be used for reducing the hardware complexity.

\[
\begin{align*}
\vec{y}_6 &= \begin{bmatrix} a & c & a & f \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b & d & e & g \\ d & -g & -b & -e \\ e & -b & g & d \\ g & -e & d & -b \end{bmatrix} \begin{bmatrix} x_0 - x_1 \\ x_1 - x_2 \\ x_2 - x_3 \\ x_3 - x_4 \end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
\vec{y}_9 &= \begin{bmatrix} a & c & a & f \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b & d & e & g \\ d & -g & -b & -e \\ e & -b & g & d \\ g & -e & d & -b \end{bmatrix} \begin{bmatrix} x_0 - x_1 \\ x_1 - x_2 \\ x_2 - x_3 \\ x_3 - x_4 \end{bmatrix}
\end{align*}
\]

In the proposed 1-D DCT/IDCT architecture, 3-bit serial distributed arithmetic technique is used for reducing processing time. Bit parallel to serial conversion for 3-bit serial distributed arithmetic operations is performed in REG of Figure 2. The architecture of 1-D DCT/IDCT unit is illustrated in Figure 4. It is modified from the architecture in [7]. 3-bit serial data are entered to 3-b adds and subs parts. Inputs are selected by connections according to matrix-vector multiplication arrangements for 1-D DCT/IDCT. 3-bit adds and subs in the front part are needed for DCT, and 18-bit adds and subtracts in the rear part are for IDCT. MUXs are used to switch the modes between DCT and IDCT. After four 3-bit serial inputs combined with even and odd pairs are entered into 1-D DCT/IDCT unit, three 4-bit addresses are generated.

The addresses are used to select coefficients in ROMs, and the selected data are accumulated. The accumulations are repeated N/K times (N = bit-width of input data, K = number of bits incoming at a time). Each accumulators can be implemented by adder tree as shown in Figure 5. Here, multipliers are substituted into ROMs and their addressings, and accumulators are implemented by carry-save adders and registers. It shows that in the first stage, 4 operands are computed, then the operands are reduced to 3, and then 3 operands are reduced to 2 operands, finally 2 operands are summed by carry propagation adder. This architecture can be easily pipelined to get high-performance, and customized as the target speed of the processor.

For matrix transposition of 1-D DCT/IDCT results, latches are used in this design, instead of the embedded RAM. It has a simple scheme and a regular structure. Especially, it can be operated at high-speed, and doesn't have complex controls. Moreover the input and output formats of the transposition can be changed according to the interface specifications. Additionally, the shape of the floor can have arbitrary dimensions. In this design, input data consist of 16-bit bundled data, while output data are changed to 3-bit serial form for reducing routing complexity.
in the input part of 1-D DCT/IDCT unit.

2.3 Timing Schedules

In this design, 1320 clocks are allocated to process a macro block, 1275 clocks are used for real processing, and 45 clocks are surplus. For the 4:2:0 chroma format, 384 pixel data consisting of six blocks within a macro block are entered by a pixel at a time. Input data and Q/IQ results are multiplexed for recursive and selective operations, and then converted from serial to parallel formats. If DCT and IDCT are processed by 3-bit serial at a time, data shortage might happen for the reason that input data are served in 8 clocks, but on the other hand they are processed within 5 clocks. Therefore, input data should be buffered for 32 clocks. For the first and second DCT/IDCT, timing schedules are illustrated in Figure 6. That is, 96 clocks are consumed for DCT and IDCT, respectively. Next input data being processed during 24(=3x8) clocks should be ready for 1-D DCT, as input data rate is insufficient as compared to processing rate.

There also exist matrix transposition delays for 5 clocks. Therefore, 29 clocks are needed to store input data in advance as illustrated in Figure 6. The first 1-D DCT/IDCT needs 5 clocks to compute each column, and the second needs 7 clocks. This is also illustrated in Figure 6. So, the total latencies to produce 2-D DCT/IDCT results are summed to 52 clocks, 40 clocks for 1-D DCT, 5 clocks for transposition, and 7 clocks for results of the first column in 2-D DCT. In the middle of processing, DCT and IDCT take 96 clocks. Finally, the latencies of Q and IQ are 3 clocks.

2.4 Interfaces of DPCM ADDER and VLC

Since timing difference between original and prediction data should be synchronized, there need memories to store the results. These memories are not redundancies due to the proposed architecture. DCT can control the time of reading pixel data from memories, accepts them block-wise. These pixel data are transformed in DCT and quantized, and then the results are stored in the memories for interfaces between DCT/Q and VLC. At this time, the data should be rearranged in zig-zag or alternate scanning order. The interface memories might be used to adjust timing schedule as well as scan conversions. As CBP(Coded Bit Pattern) information in the coded macro block data should be investigated at the previous macro block, DCT/Q compression procedures are executed in the earlier macro block period than VLC. If the dual-port memory for the interface is used, it can also serve the data for inverse quantization and 2-D IDCT. DCT/Q results are written into the memory in scanning order, and then VLC and IQ/IDCT read the input data in raster and scanning order, respectively.

III. DESIGN AND IMPLEMENTATION RESULTS

In this paper, we proposed an area efficient scheme for transform coding module in MPEG-2 video encoder. The design shows area efficiency with regard to existing architectures in the case that they are used for MPEG-2 video encoder.
Table I. Accuracy simulation results of the proposed architecture

<table>
<thead>
<tr>
<th>Items</th>
<th>ITU-T H.261</th>
<th>proposed design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel Peak Error</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Peak Mean Square Error</td>
<td>0.06</td>
<td>0.052</td>
</tr>
<tr>
<td>Overall Mean Square Error</td>
<td>0.02</td>
<td>0.007</td>
</tr>
<tr>
<td>Peak pixel Mean Error</td>
<td>0.015</td>
<td>0.0034</td>
</tr>
<tr>
<td>Overall Mean Error</td>
<td>0.0015</td>
<td>0.00017</td>
</tr>
</tbody>
</table>

Table II. Gate counts of the proposed architecture

<table>
<thead>
<tr>
<th>Components</th>
<th>Transistor Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-D DCT/IDCT</td>
<td>94876</td>
</tr>
<tr>
<td>Controller</td>
<td>6432</td>
</tr>
<tr>
<td>Q/I/Q</td>
<td>25396</td>
</tr>
<tr>
<td>SAG</td>
<td>3920</td>
</tr>
<tr>
<td>Total</td>
<td>130624</td>
</tr>
</tbody>
</table>

Table III. Characteristics of the proposed design

<table>
<thead>
<tr>
<th>Functions</th>
<th>Two-dimensional DCT</th>
<th>Two-dimensional IDCT Quantization Inverse Quantization Zigzag/alternate scanning</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of transistors</td>
<td>130624</td>
<td></td>
</tr>
<tr>
<td>Technology</td>
<td>0.5 μm CMOS, TLM, 3.3 V</td>
<td></td>
</tr>
<tr>
<td>Core size</td>
<td>9.76 mm²</td>
<td></td>
</tr>
<tr>
<td>Clock rate</td>
<td>80 MHz</td>
<td></td>
</tr>
<tr>
<td>Internal wordlength</td>
<td>22-bit</td>
<td></td>
</tr>
<tr>
<td>Accuracy</td>
<td>ITU-T H.261 compatible</td>
<td></td>
</tr>
</tbody>
</table>

The implementation results are summarized as shown in Table I and II. In Table I, the accuracy simulation results of the proposed architecture are compared with the specifications in ITU-T H.261 [8]. It shows that our design meets the specifications sufficiently. Table 2 describes the hardware complexity of the design. According to the results, our design consumes about 24% and 50% less transistors than the existing architectures [5,6]. In [5] and [6], 102 K and 67 K transistors were consumed, respectively. If these are used for MPEG-2 video encoder, 204 K and 134 K transistors should be consumed roughly. But in our design, only 101.3 K transistors are used for both of the 2-D DCT and 2-D IDCT. Table III shows the characteristics of the proposed design. The core area of the whole transform coding module is measured to 9.76 mm² in 0.5 μm CMOS, TLM, 3.3 V technology. The critical path delay of this design is 12 ns in ROM-ACCU of 1-D DCT/IDCT unit. Figure 7 shows the core layout photograph of the proposed architecture.

As a result, the proposed architecture carries advantages of reducing hardware complexity. Even though the existing architectures are adjusted to have low-complexity at the expense of speed degrading, the proposed architecture is more efficient than the existing architectures in [5] and [6] in the applications like MPEG-2 encoder using two 2-D DCT/IDCT units.

IV. CONCLUSIONS

This kind of attempt has not been recognized as an efficient method for implementing MPEG-2 video encoder among hardware designers. But in this paper, the proposed architecture of transform coding module is shown as an area efficient method. Though the technology used in this design is different from [5] and [6], the transistor counts are not influenced by technology. So, the transistor counts and their comparisons between our design and the existing designs are reasonable. This design was synthesized and verified by high-level CAD tools. By using the netlists extracting from the synthesis tool, the layout was performed by the back-end tool. Post-layout simulation was performed again extracting netlists including routing as well as delay information from the layout results. For the above works, 0.5 μm, CMOS, TLM, 3.3 V standard cell libraries are used. According to the results of this design, the utilization of hardware resources is enhanced. As a result, this architecture can be characterized to maximize the utilization of the hardware resources. This method can be utilized for the encoders having similar structure as the MPEG-2 video encoder. It also can be applied to the ASIC chips for multimedia services especially requiring low hardware complexity.

REFERENCES


ACKNOWLEDGEMENTS

The authors wish to thank Kyung-yuk Min in Hanyang University for helping high-level functional verification and lots of useful discussion.