ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.2, Pages : 85-91 (2013)
Special Issue of NCRTECE 2013 - Held during 8-9 February, 2013 in SMK Fomra Institute of Technology, OMR, Thaiyur, Kelambakkam, Chennai

SPACE TIME SCHEDULING STRATEGY FOR EFFICIENT 2-D DCT ARCHITECTURE

[1] Kiruthika P., [2] Sivanthiram C.S. .,
[1] M.E- Student VelTech Multitech Dr.Rangarajan Dr.Sakunthala Engg College
E-mail:kiruthibala11@gmail.com, Phone No: 9677215569
[2] Asst Professor VelTech Multitech Dr.Rangarajan Dr.Sakunthala Engg College
E-mail:sivanthiram024@gmail.com, Phone No: 9944216430

Abstract

Distributed arithmetic (DA) has been generally used to apply inner product calculation with a fixed input. Conventional ROM-based DA suffers from large ROM requirements. A new DA algorithm is used to expand the fixed input as an alternative of the variable input into bit level as in ROM-based DA. Thus the new DA algorithm can take advantage of shared partial sum-of- products and sparse non zero bits in the fixed input to reduce the number of computation. Unlike ROM based DA that stores the pre computed results. In this paper, a spatial and time scheduling strategy, called the space-time scheduling (STS) strategy that achieves high image resolutions in real-time systems is proposed. The proposed spatial scheduling strategy includes the capacity to select the distributed arithmetic (DA)-precision bit length as preferred as 9 bits as an alternative of the usual 12 bits based on test image simulations, a hardware sharing architecture that reduces the hardware cost, and the proposed time scheduling strategy arranges different dimensional computations in that it can calculate 1-D and 2-D transformations at the same time in single 1-D discrete cosine transform (DCT) core.

Keywords: Discrete cosine transform (DCT), Binary signed-digit (BSD), space-time scheduling (STS).

I. INTRODUCTION

1.1 Definition of DCT

DISCRETE COSINE TRANSFORM (DCT) is a widely used transform engine for image and video compression applications. In recent years, the development of visual media has been progressed towards high-resolution specifications, such as high definition television (HDTV). Consequently, a high-accuracy and high-throughput rate component is needed to meet future specifications. In addition, in order to reduce the manufacturing costs of the integrated circuit (IC), a low hardware cost design is also required. Therefore, a high performance video transform engine that utilizes high accuracy, a small area, and a high-throughput rate is desired for VLSI Designs.

The 2-D DCT core design has often been implemented using either direct [2]–[4] or indirect methods. The direct methods include fast algorithms that reduce the computation complexity by mapping and rotating the DCT transform into a complex number. However, the structure of the DCT is not as regular as that of the fast Fourier transform (FFT). A

regular 2-D DCT core using the direct method that derives the 2-D shifted FFT (SFFT) from the 2-D DCT algorithm format, and shares the hardware in FFT/IFFT/2-D DCT computation is implemented.

On the other hand, the 2-D DCT cores using the indirect method are implemented based on transpose memory and have the following two structures: 1) two 1-D DCT cores and one transpose memory (TMEM) and 2) a single 1-D DCT core andone TMEM.

In the first structure, the 2-D DCT core has a high throughput rate, because the two 1-D DCT cores compute the transformation simultaneously. In order to reduce the area overhead, a single 1-D DCT core is applied in the second structure, thereby saving hardware costs. . However, the additional input buffers are needed in order to temporarily store the input data during the 2nd-D transformation. Madisetti *et al.* present a hardware sharing technique to implement the 2-D DCT core using a single 1-D

DCT core . The 1-D DCT core can calculate the 1st-D and 2nd-D DCT computations simultaneously, and the throughput achieves 100 Mpels/s.

As a result, it is obvious that a tradeoff is required between the hardware cost and the speed. Tumeo *et al.* find a balance point between the area required and the speed for 2-D DCT designs and present a multiplier-based pipeline fast 2-D DCT accelerator implemented using a field-programmable gate arrays (FPGA) platform . Additionally, several 2-D DCT designs are implemented based on FPGA for fast verification.

In this paper, an 8 8 2-D DCT core that consists of a single 1-D DCT core and one TMEM is proposed using a strategy known as space-time scheduling (STS). Due to the accuracy simulations in DA-based binary signed-digit (BSD) expression, 9-bit DA precision is chosen in order to meet the requirements of the peak-signal-to-noise-ratio (PSNR) outlined in previous works . Furthermore, the proposed DCT core is designed based on a hardware sharing architecture with BSD DA-based computation so as to reduce the area cost. The arithmetics share the hardware resources during the four time slots in the DCT core design. A 100% hardware utilization is also achieved by using the proposed time scheduling strategy, and the 1st-D and 2nd-D DCT computations can be calculated at the same time. Therefore, a high performance transform engine with high accuracy, small area, and a high-throughput rate has been achieved. This paper is organized as follows, the mathematical derivation of the BSD format distributed arithmetic is given. The proposed 8* 8 2-D DCT architecture that includes an analysis of the coefficient bits, the hardware sharing architecture, and the proposed timing scheduling strategy.

II. Mathematical Derivation of BSD Format Distributed Arithmetic

The inner product for a general matrix multiplication-and accumulation can be written as follows:

$$Y = \mathbf{A}^T\mathbf{X} = \sum_{i=1}^{L} A_i X_i \qquad \textbf{(1)}$$

as, Where A is an N-BSD number ,the above eqn is expresses

$$Y = \begin{bmatrix} 2^0 & 2^{-1} & \cdots & 2^{-(N-1)} \end{bmatrix} \cdot \begin{bmatrix} A_{1,0} & A_{2,0} & \cdots & A_{L,0} \\ A_{1,1} & A_{2,1} & \cdots & A_{L,1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{1,(N-1)} & A_{2,(N-1)} & \cdots & A_{L,(N-1)} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_L \end{bmatrix}$$

$$= \begin{bmatrix} 2^0 & 2^{-1} & \cdots & 2^{-(N-1)} \end{bmatrix} \begin{bmatrix} Y_0 \\ Y_1 \\ \vdots \\ Y_{(N-1)} \end{bmatrix} \qquad (2)$$

The matrix is called the DA coefficient matrix. In (2), can be calculated by adding or subtracting x. With A not equal to 0 , and then the transform output Y can be obtained by shifting and adding every non-zero . Thus, the inner product computation can be implemented by using shifters and adders instead of multipliers. Therefore, a smaller area can be achieved by using BSD DA-based architecture.

3. PROPOSED 8 8 2-D DCT CORE DESIGN

A spatial and time scheduling strategy, called the space-time scheduling (STS) strategy that achieves high image resolutions in real-time systems is proposed. The proposed spatial scheduling strategy includes the ability to choose the distributed arithmetic (DA)-precision bit length, a hardware sharing architecture that reduces the hardware cost, and the proposed time scheduling strategy arranges different dimensional computations in that it can calculate first-dimensional and second-dimensional transformations simultaneously in single 1-D discrete cosine transform (DCT) core to reach a hardware utilization of 100%. The DA-precision bit length is chosen as 9 bits instead of the traditional 12 bits based on test image simulations. In addition, the proposed hardware sharing architecture employs a binary signed-digit DA architecture that enables the arithmetic resources to be shared during the four time slots.

In this paper we present the design of a reconfigurable transformation IP core which can perform the multiplier less 1- D 8-point DCT and the 1-D 8-point/4-point integer transforms for multi-standard Video CODECs by reusing the architecture of our previous CORDIC based Loffler DCT (CLDCT) In this work we have successfully implemented a low-power and high-quality CLDCT which not only reduces the computational complexity from 11 multiplier and 29 add operations to 38 add and 16 shift operations but also obtains a transformation quality as good as the original Loffler DCT.

Here, we present the integration of 8-point/4-point integer transforms in this CLDCT, such that a reconfigurable architecture for Discrete Cosine and Integer Transform (DCIT) is obtained, which can be used for multi-functional SOC designs. Even if the 17 Multiplier could be replaced by Canonical Signed Digit (CSD) representation; the QDCT and NQDCT still need more than 300 add operations in the worst case. In order to realize an efficient QDCT architecture, we present a CORDIC Scalar (C-Scalar) by involving limited CORDIC compensation iterations. This C-Scalar has four stages and requires 8 add and 10 shift operations.

The combination of two 1-D DCITs, a row-column transposition memory and four C-Scalars forms a 2-D Quantized DCIT (QDCIT) which requires only 120 add and 80 shift operations. The final architecture can not only perform multiplier less 8×8 QDCT but contain reconfigurable modules such that it can also perform 8×8/4×4 quantized integer transforms. Furthermore, it also retains the good transformation quality compared to the Loffler DCT in terms of PSNR results. Applications include portable video, wireless communications, and sensor data processing.

This paper focuses on two novel aspects of low-power DSP system design: Exploitation of correlation and reduced data dynamic range for switching activity reduction and precision reduction for computation related to perceptually insignificant data. The case study presented in this paper is a low-power discrete cosine transform (DCT) core processor with direct applications to video and still-image compression systems (MPEG, JPEG). The main design ideas can be adapted to other DSP contexts where the input data exhibit substantial correlation or reduced dynamic range. This section introduces the proposed 8 ×8 2-D DCT core implementation.

The 2-D DCT is defined as

$$Y_{u,v} = \frac{1}{4} k_u k_v \sum_{i=0}^{7} \sum_{j=0}^{7} x_{i,j} \cos\left(\frac{(2i+1)u\pi}{16}\right) \times \cos\left(\frac{(2j+1)v\pi}{16}\right) \qquad (3)$$

Consider the 1-D 8-point DCT first

$$Z_n = \frac{1}{2} k_n \sum_{m=0}^{7} x_m \times \cos\left(\frac{(2m+1)n\pi}{16}\right) \qquad (4)$$

By neglecting the scaling factor 1/2, the 1-D 8-point DCT in (4) can be divided into even and odd parts,and , as listed in (5) and (6), respectively.

$$\mathbf{Z}_e = \begin{bmatrix} Z_0 \\ Z_2 \\ Z_4 \\ Z_6 \end{bmatrix} = \begin{bmatrix} c_4 & c_4 & c_4 & c_4 \\ c_2 & c_6 & -c_6 & -c_2 \\ c_4 & -c_4 & -c_4 & c_4 \\ c_6 & -c_2 & c_2 & -c_6 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \mathbf{C}_e \cdot \mathbf{a} \qquad (5)$$

$$\mathbf{Z}_o = \begin{bmatrix} Z_1 \\ Z_3 \\ Z_5 \\ Z_7 \end{bmatrix} = \begin{bmatrix} c_1 & c_3 & c_5 & c_7 \\ c_3 & -c_7 & -c_1 & -c_5 \\ c_5 & -c_1 & c_7 & c_3 \\ c_7 & -c_5 & c_3 & -c_1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \mathbf{C}_o \cdot \mathbf{b} \qquad (6)$$

where $c_i = \cos(i\pi/16)$ and

$$\mathbf{C}_e = \begin{bmatrix} c_4 & c_4 & c_4 & c_4 \\ c_2 & c_6 & -c_6 & -c_2 \\ c_4 & -c_4 & -c_4 & c_4 \\ c_6 & -c_2 & c_2 & -c_6 \end{bmatrix} \qquad (7)$$

$$\mathbf{C}_o = \begin{bmatrix} c_1 & c_3 & c_5 & c_7 \\ c_3 & -c_7 & -c_1 & -c_5 \\ c_5 & -c_1 & c_7 & c_3 \\ c_7 & -c_5 & c_3 & -c_1 \end{bmatrix} \qquad (8)$$

$$a = \begin{bmatrix} x_0 + x_7 \\ x_1 + x_6 \\ x_2 + x_5 \\ x_3 + x_4 \end{bmatrix}, \quad b = \begin{bmatrix} x_0 - x_7 \\ x_1 - x_6 \\ x_2 - x_5 \\ x_3 - x_4 \end{bmatrix}. \qquad (9)$$

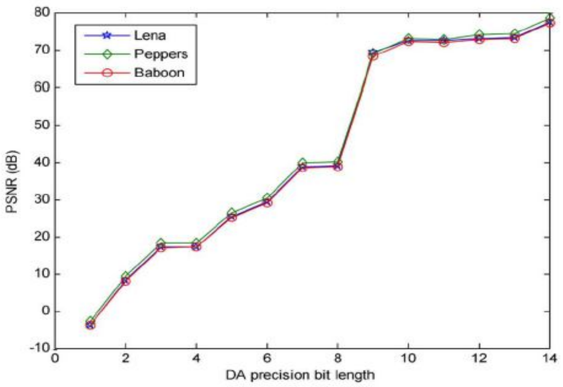
For the 2-D DCT core implementation, the three main strategies for increasing the hardware and time utilization, called the space-time scheduling (STS) strategy, are proposed and listed as follows:

- find the DA-precision bit length for the BSD representation
- to achieve system accuracy requirements;
- share the hardware resource in time to reduce area cost;
- plan an 100% hardware utilization by using time scheduling strategy.

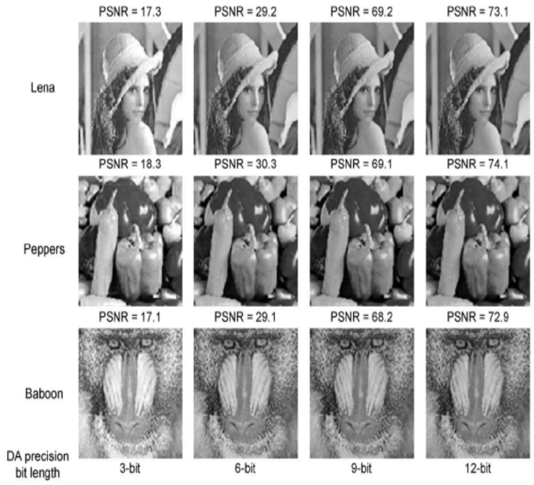
3.1 Analysis of the Coefficient Bits

The seven internal coefficients from to for the 2-D DCT transformation are expressed as BSD representations in order to save computation time and hardware cost, as well as to achieve the requirements of PSNR. The system PSNR is defined [1] as follows:

$$\mathrm{PSNR} = 10\log_{10}\frac{255^2}{\mathrm{MSE}_I} \qquad (10)$$



(a)



(b)

Fig. 1 Simulation for system PSNR in different BSD bit length expressions.

3.2. Hardware Sharing Strategy

All modules in the 1-D DCT core, including the modified two-input butterfly (MBF2), the pre-reorder, the process element even (PEE), the process element odd (PEO), and the postreorder share the hardware resources in order to reduce the area cost. Moreover, the 8 8 2-D DCT core is implemented using a single 1-D DCT core and one TMEM. The architecture of the 2-D DCT core is described in the following sections.

3.2.1 MBF2 Architecture:

Modified Butterfly Module is easily implemented using a two-input butterfly module called BF2. In general, the BF2 has a hardware utilization rate in the adder and subtracter of 50%. In order to enable the hardware resources to be shared, additional multiplexers and Reorder Registers are added to the proposed MBF2 module. The Reorder Registers consist of four word registers that use the control signals to select the input data and use enable signals to output or hold the data. Similar to BF2, the operation of the proposed MBF2 has an eight-clock-cycle period. In the first four cycles, the 1st-D input data (x0, x1, x2, and x3) shift into Reorder Registers 1, and the 2nd-D input Z' data execute the operation of addition and subtraction. In the next four cycles, the operations of 1st-D and 2nd-D will be changed.

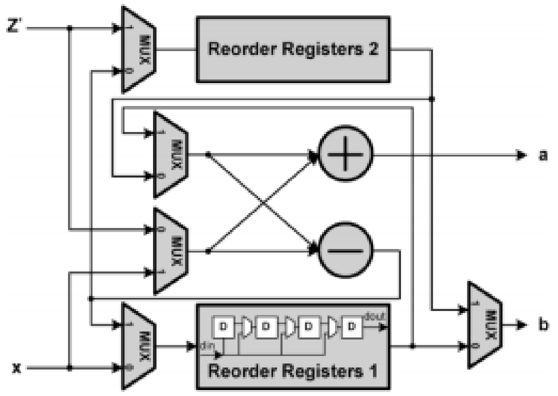


Fig.2 MBF2 Architecture

3.2.2 Processing Element:

DA-based computation, the even part and odd part transformations can be implemented using PEE and PEO, respectively. The even part transformation can be expanded for the DA-based computation formats and can share the hardware resources at the bit level. The coefficient vector has two combinations of [C4C4C4C4] and [C2C6C6C2] for (Z0, Z4) and (Z2, Z6) transform outputs, respectively. Using given input data et,0, et,1, et,2 , and et,3 , the transform Output Ze needs only three adders, a Data-Distributed module, and one even part adder-tree (EAT) to obtain the result for Ze during the four time slots and the EAT sums the different weighted values in tree-like adders to complete the transform output Ze . Similarly, the odd part transformation it can be implemented in a DA-based format using four adders and one odd part adder-tree (OAT). The EAT and OAT can be implemented using the error-compensated adder tree to improve the computation accuracy. Moreover, there are three pipeline stages (two in both the EAT and the OAT) in each PEE and PEO module that enable high speed computation to be achieved. Inputs enter into the processing element from the upper left. The first step is for each of these inputs to be multiplied by their respective weighting factor (w(n)). Then these modified inputs are fed into the summing function, which usually just sums these products. Yet, many different types of operations can be selected. These operations could produce a number of different values which are then propagated forward; values such as the average, the largest, the smallest, the OR -ed values, the AND -ed values, etc. Furthermore, most commercial development products allow software engineers to create their own summing functions via routines coded in a higher level language (C is commonly supported). Sometimes the summing function is further complicated by the addition of an activation function which enables the summing function to operate in a time sensitive way. Either way, the output of the summing function is then sent into a transfer function. This function then turns this number into a real output via some algorithm. It is this algorithm that takes the input and turns it into a zero or a one, a minus one or a one, or some other number. The transfer functions that commonly supported are sigmoid, sine, hyperbolic tangent, etc. This transfer function also can scale the output or control its value via thresholds. The result of the transfer function is usually the direct output of the processing element.

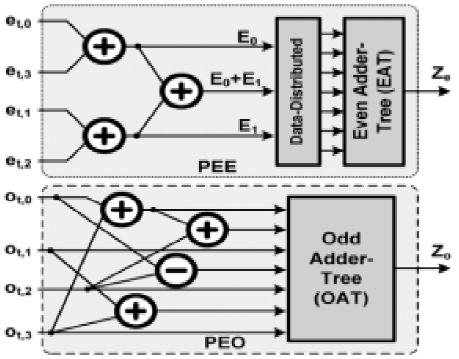


Fig .3 PEE and PEO Architecture

3.2.3 Post Reorder:

The data sequence after the PEE and PEO must be merged and re-permuted in the Post-Reorder module. The order of the original 8-point data sequence from the PEE and the PEO is {Z0, Z2, Z4, Z6, Z1, Z3, Z5, and Z7}. After the Post-Reorder module permutes the data order, the data sequence is ordered as {Z0, Z2, Z4, Z6, Z1, Z3, Z5, Z7 }. Two multiplexers select the data that is fed into the different Reorder Registers in order to permute the output order. Z is the transform output for the 1st-D DCT that will input into the TMEM. In addition, the 2nd-D DCT transform output is completed after the permutation by Reorder Registers 4.

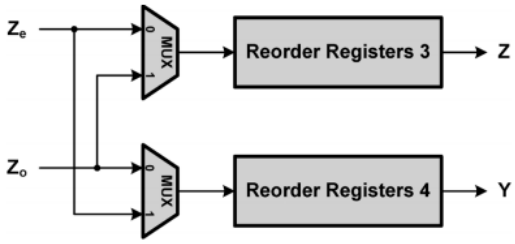


Fig .4 Post-Reorder Architecture

3.2.4 TMEM:

The TMEM is implemented using 64-word 12-bit dual-port registers and has a latency of 52 cycles. Based on the time scheduling strategy and result of the time scheduling strategy, the 1st-D and 2nd-D transforms are able to be computed simultaneously.

2-D DCT Core Architecture:

To save hardware costs, the proposed 2-D DCT core, is implemented using a single 1-D DCT core and one TMEM. The 1-D DCT core includes an MBF2, a Pre-Reorder module, a PEE, a PEO, a Post-Reorder module, and one TMEM. The TMEM is implemented using 64-word 12-bit dual-port registers and has a latency of 52 cycles. Based on the time scheduling strategy, a hardware utilization of 100% can be achieved.

4 Conclusion

The proposed spatial scheduling strategy includes the capacity to select the distributed arithmetic (DA)-precision bit length, a hardware sharing architecture that reduces the hardware cost, and the proposed time scheduling strategy arranges different dimensional computations in that it can calculate 1-D and 2-D transformations at the same time in single 1-D discrete cosine transform (DCT) core. The structure of Re-order Register is customized and designed and its output is obtained. The DA-precision bit length is preferred as 9 bits as an alternative of the usual 12 bits based on test image simulations. the reorder register is implemented in the 2-D DCT architecture and the area is reduced.

REFERENCES

[1] A. M. Shams, A. Chidanandan, W. Pan, and M. A. Bayoumi, “NEDA: Alow-power high-performance DCT architecture,” IEEE Trans. Signal Process., vol. 54, no. 3, pp. 955–964, Mar. 2006.

[2] A. Madisetti and A. N. Willson, “A 100 MHz 2-D 8 X 8 DCT/IDCT processor for HDTV applications,” IEEE Trans. Circuits Syst. Video Technol., vol. 5, no. 2, pp. 158–165, Apr. 1995.

[3] A. Tumeo, M. Monchiero, G. Palermo, F. Ferrandi, and D. Sciuto, “A pipelined fast 2DDCT accelerator for FPGA-based SOCs,” in Proc. IEEE Comput. Soc. Annu. Symp. VLSI., 2007, pp. 331–336.

[4] C. C. Sun, P. Donner, and J. Gotze, “Low-complexity multi-purpose IP core for quantized discrete cosine and integer transform,” in Proc. IEEE Int. Symp. Circuits Syst.,2009, pp. 3014–3017.

[5] C. Peng, X. Cao, D. Yu, and X. Zhang, “A 250 MHz optimized distributed architecture of 2D 8 x 8 DCT,” in Proc. Int. Conf. ASIC, 2007, pp. 189–192.

[6] C. T. Lin, Y. C. Yu, and L. D. Van, “Cost-effective triple-mode reconfigurable pipeline FFT/IFFT/2-D DCT processor,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 16, no. 8, pp. 1058–1071, Aug. 2008.

[7] C. Y. Huang, L. F. Chen, and Y. K. Lai, “A high-speed 2-D transform architecture with unique kernel for multi-standard video applications,” in Proc. IEEE Int. Symp. Circuits Syst., 2008, pp. 21–24.39

[8] E. Feig and S.Winograd, “Fast algorithms for the discrete cosine transform,” IEEE Trans. Signal Process., vol. 40, no. 9, pp. 2174–2193, Sep. 1992.

[9] H. C. Hsu, K. B. Lee, N. Y. Chang, and T. S. Chang, “Architecture design of shapeadaptive discrete cosine transform and its inverse for MPEG-4 video coding,” IEEE Trans. Circuits Syst. Video Technol., vol. 18, no. 3, pp. 375–386, Mar. 2008.

[10] J. I. Guo, R. C. Ju, and J. W. Chen, “An efficient 2-D DCT/IDCT core design using cyclic convolution and adder-based realization,” IEEE Trans. Circuits Syst. Video Technol., vol. 14, no. 4, pp. 416–428, Apr. 2004.

[11] M. Alam, W. Badawy, and G. Jullien, “A new time distributed DCT architecture for MPEG-4

hardware reference model," IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 5, pp. 726–730, May 2005.

[12] S. C. Hsia and S. H.Wang, "Shift-register-based data transposition for cost-effective discrete cosine transform," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 15, no. 6, pp. 725–728, Jun. 2007.

[13] T. Xanthopoulos and A. P. Chandrakasan, "A low-power DCT core using adaptive bitwidth and arithmetic activity exploiting signal correlations and quantization," IEEE J. Solid-State Circuits, vol. 35, no. 5, pp. 740–750, May 2000.

[14] Y. P. Lee, T. H. Chen, L. G. Chen, M. J. Chen, and C. W. Ku, "A cost-effective architecture for 8x8 two-dimensional DCT/IDCT using direct method," IEEE Trans. Circuits Syst. Video Technol., vol. 7, no. 3, pp. 459–467, Jun. 1997.

[15] Y.Wang, J. Ostermann, and Y. Zhang, Video Processing and Communications, 1st ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.