A High Speed FPGA Implementation of the 2D DCT for Ultra High Definition Video Coding

Paris Kitsos and Nikolaos S. Voros
Department of Telecommunication Systems and Networks
Technological Educational Institute of Messolonghi
Branch of Nafpaktos, Greece
e-mail: {pkitsos, voros}@teimes.gr

Tasos Dagiuklas
Department of Computer Science
Hellenic Open University
Patras, Greece
e-mail: dagiuklas@eap.gr

Athanassios N. Skodras
Department of Electrical and Computer Engineering
University of Patras
Patras, Greece
e-mail: skodras@upatras.gr

Abstract—**This paper presents two high performance FPGA architectures for the 2D DCT computation for Ultra High Definition video coding systems. Both architectures use Distributed Arithmetic to perform the necessary multiplications instead of traditional multipliers. The first architecture uses 105 clock cycles to transform an 8x8 block and reaches a rate of up to 206 samples per second at a 338.5 MHz frequency, while the second one requires 65 cycles for each 8x8 block and achieves a rate equal to 252 samples per second at 256 MHz. Both architectures have been implemented using VHDL. Virtex7 FPGA of Xilinx has been used for the realization of both implementations.**

Keywords—Video Coding, 2D DCT, Distributed Arithmetic, FPGA Implementation, VHDL.

I. INTRODUCTION

Digital video has become mainstream and is being used in a wide range of applications including DVD, digital TV, HDTV and video telephony. These applications are feasible because of the advances in computing and communication technologies as well as efficient video compression algorithms [1]. Video coding plays an important role in these applications, as it provides the necessary data compression to allow the transmission and storage of digital video contents in the currently available facilities and networks. However, with the increasing presence of high and ultra high definition video contents, the standardization communities have started the process to change the current video coding standard (H.264/AVC [2]) with a new one called High Efficiency Video Coding (HEVC) [3] targeting the reduction of the coding rates in 50% for the same quality.

The Discrete Cosine Transform (DCT) is used for transform coding in state of the art video codecs such as H.264. Usually, the DCT is implemented through two 1D transforms, one along the rows and the other along the columns. Hardware implementations are of special interest for the realization of highly parallel algorithms that can achieve much higher throughput (data rate) than software solutions. In addition, special purpose DCT hardware discharges the computational load from the processor and therefore improves the performance of the multimedia system. The data rate is directly influencing the quality-of-experience of multimedia content. In order to improve the data rate, multiplierless techniques are employed, which involve the use of memories (RAMs, ROMs) or Look-Up Tables (LUTs) to store pre-computed values of coefficient operations. A well known technique is Distributed Arithmetic (DA) [4], which appeared as a very efficient solution especially suited for LUT-based FPGA architectures.

In this paper two efficient 2D DCT architectures are presented, capable to manage 8x8 image blocks. Both architectures use DA in order to improve the time performance. The architecture of 2D DCT in [5] was used as a reference architecture. Although this architecture was optimized for low power applications and implemented in ASIC, there is no need for high level of throughput, since two high speed FPGA architectures are implemented in this paper.

This paper is organized as follows: In Section II the DCT algorithm is briefly described. The proposed architectures and hardware implementations are presented in detail in Section III. The synthesis results and comparison evaluation for the FPGA implementation are shown in Section IV, and conclusions are drawn in Section V.

II. 2D DCT ALGORITHM

The 2D DCT architecture uses the row–column distributed arithmetic version of the Chen [6] fast DCT algorithm. The first step of the Chen algorithm is a factorization of the DCT-II [7] matrix such that the subsequent computation of the even indexed coefficients are fully separated from the computation of the odd indexed coefficients. The 1D DCT coefficients X_k, k=0,1,…,7 for an 8-point input vector x_n, n=0,1,…,7 can be expressed as follows:

$$\begin{bmatrix} X0 \\ X2 \\ X4 \\ X6 \end{bmatrix} = \begin{bmatrix} A & A & A & A \\ B & C & -C & -B \\ A & -A & -A & A \\ C & -B & -B & C \end{bmatrix} \begin{bmatrix} x0+x7 \\ x1+x6 \\ x2+x5 \\ x3+x4 \end{bmatrix} \quad (1)$$

where $A = \cos(\frac{\pi}{4})$, $B = \cos(\frac{\pi}{8})$, $C = \sin(\frac{\pi}{8})$, $D = \cos(\frac{\pi}{16})$,

$E = \cos(\frac{3\pi}{16})$, $F = \sin(\frac{3\pi}{16})$ and $G = \sin(\frac{\pi}{16})$.

$$\begin{bmatrix} X1 \\ X3 \\ X5 \\ X7 \end{bmatrix} = \begin{bmatrix} D & E & F & G \\ E & -G & -D & -F \\ F & -D & G & E \\ G & -F & E & -D \end{bmatrix} \begin{bmatrix} x0-x7 \\ x1-x6 \\ x2-x5 \\ x3-x4 \end{bmatrix} \quad (2)$$

III. HARDWARE ARCHITECTURE

The hardware architecture of the 2D DCT is shown in Fig. 1. The design has a 64-bit data input and 112-bit output. Each input coefficient is equal to 8 bits. So, the eight coefficients (64-bit) of each row are shifted into the register during the first clock cycle.
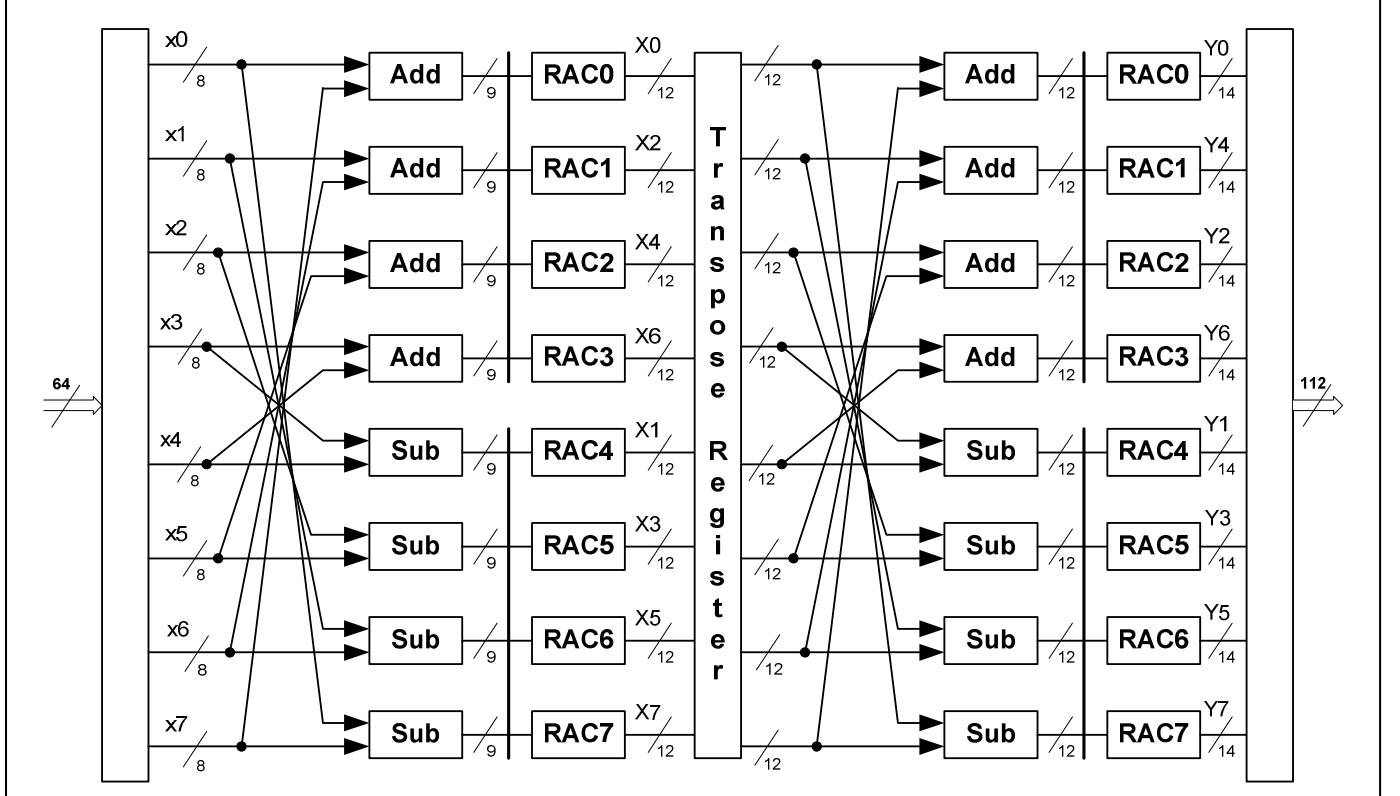

Figure 1.   The 8x8 2D DCT architecture

In the next stage, the adders and subtractors perform the first butterflies. In order to keep full accuracy, the outputs of the butterflies should be 9 bits long. Then, the data are loaded into parallel-in serial-out registers that repackage the data into 4-bit addresses that serially feed the ROM and Accumulators, from RAC0 to RAC7, with MSB first.

A straightforward architecture of the RAC by using DA structure is depicted in Fig. 2. Although the serial inputs limit the performance of such a structure a better performance can be obtained by using more hardware resources.
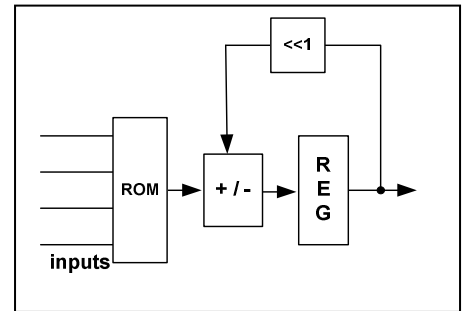

Figure 2.   Straightforword RAC architecture.

Fig. 3a illustrates two bit-sum that can be computed at a time by duplicating the ROM and adder tree. The first ROM is used by even indexed input bits and the second one is used by odd indexed input bits. The odd bit partials are left shifted to properly weight the result and added to the even partials before accumulating the aggregate. Since two bits are taken at a time, the scaling accumulator has to shift the feedback by two bits on the left. The ROM is made 10-bit wide to accommodate for the largest sum without overflow. ROMs contain the sums of the constant coefficients, for all the possible serial input combinations as shown in Fig. 3b. The adder that sums the odd and even partials is 10-bit and the accumulator adder is 20-bit long to accommodate for the continuously increasing operands due to the logical shifting in the feedback path. The results of the first RACs computations are 20-bit 2's-complement and are rounded to 12-bit 2's-complement before being stored in the transpose register. Actually, the results of the first RACs are the row 1D DCT. The resulted coefficients are taken in the order of X0, X2, X4, X6, X1, X3, X5, X7. Every 5 cycles, the final results within each RAC are computed. As it has been mentioned above, two bits per cycle are processed in the RAC. So, in order to balance the outputs of the butterflies, which are 9 bits long, a bit equal to zero is appended at the end of the butterflies' outputs. This does not cost any inaccuracy because the value of the ROM zero address is equal to zero.
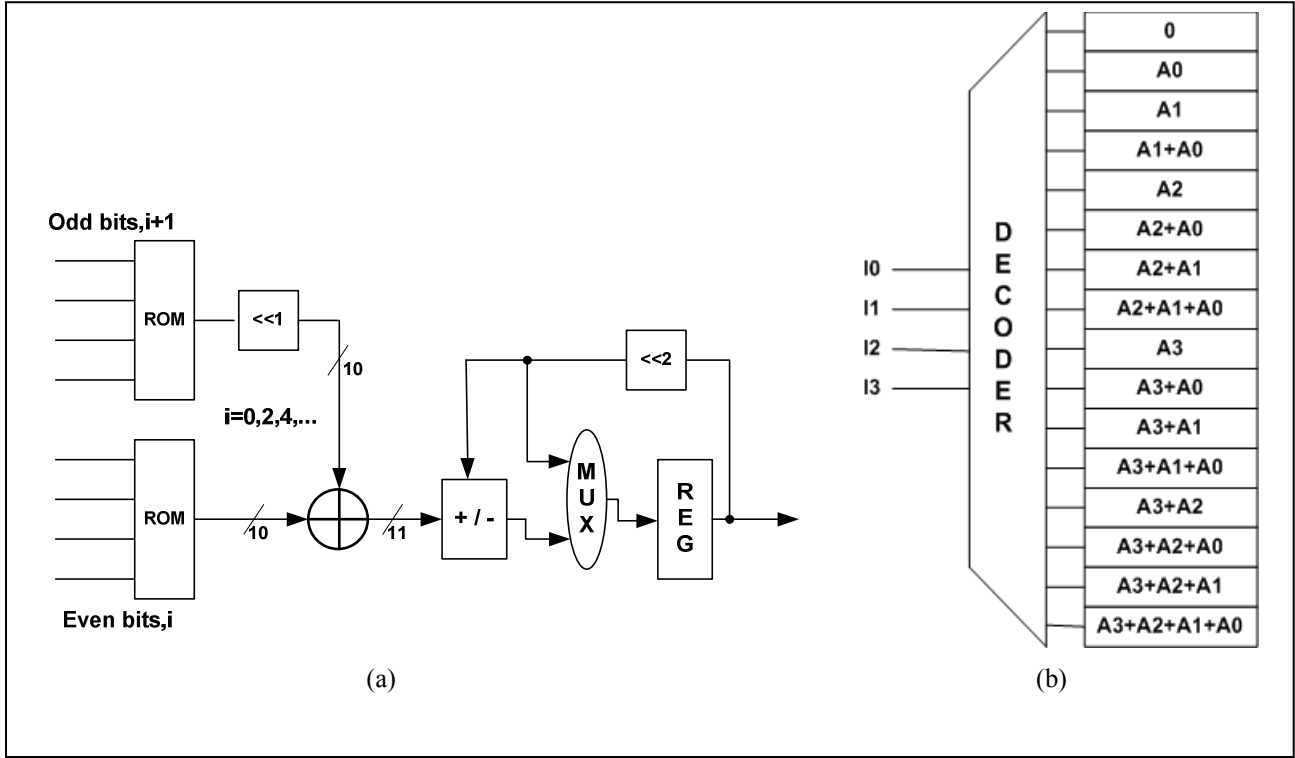

Figure 3.   a) The proposed RAC architecture, b) The content of the ROM

The transpose register consists of eight 96-bit serial-input parallel-output registers and eight 8-to-1 multiplexers. The architecture of the transpose register is depicted in Fig. 4. So, the coefficients that have been computed by the first RACs are rearranged in an ascending order and stored in each register. After 8 executions the transpose register has been filled with the DCT coefficients of the eight rows. Then, each register outputs its content in parallel and enables the eight multiplexers. During the first operation the multiplexers select the most significant byte from each register and feed the second 1D DCT with the first column. During the second operation, the multiplexers feed the second column and so on.

Then, a similar 1D DCT is applied on the eight columns. As shown in Fig. 1, in order to meet high levels of accuracy, the outputs of the butterflies are 12 bits long and the outputs of the RACs are 14 bits long.
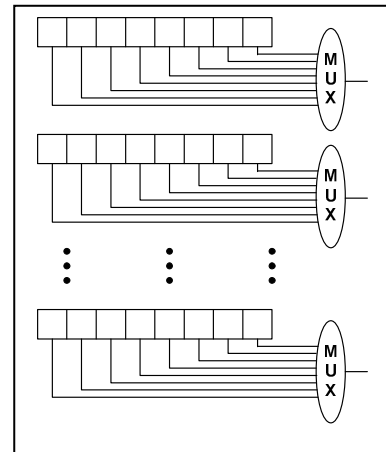

Figure 4.   The Transpose Register Architecture

The final results within each RAC are computed every 6 cycles. This architecture requires a maximum transpose register of 768 1-bit registers and parallelized row and column RAC stages. Additionally, this architecture needs 105 clock cycles in order to process an 8x8 image block.

In order to increase the time performance of the architecture, a second option of the RAC architecture is proposed. Specifically, four bit-sums can be computed in parallel using four ROMs. This architecture is illustrated in Fig. 5.
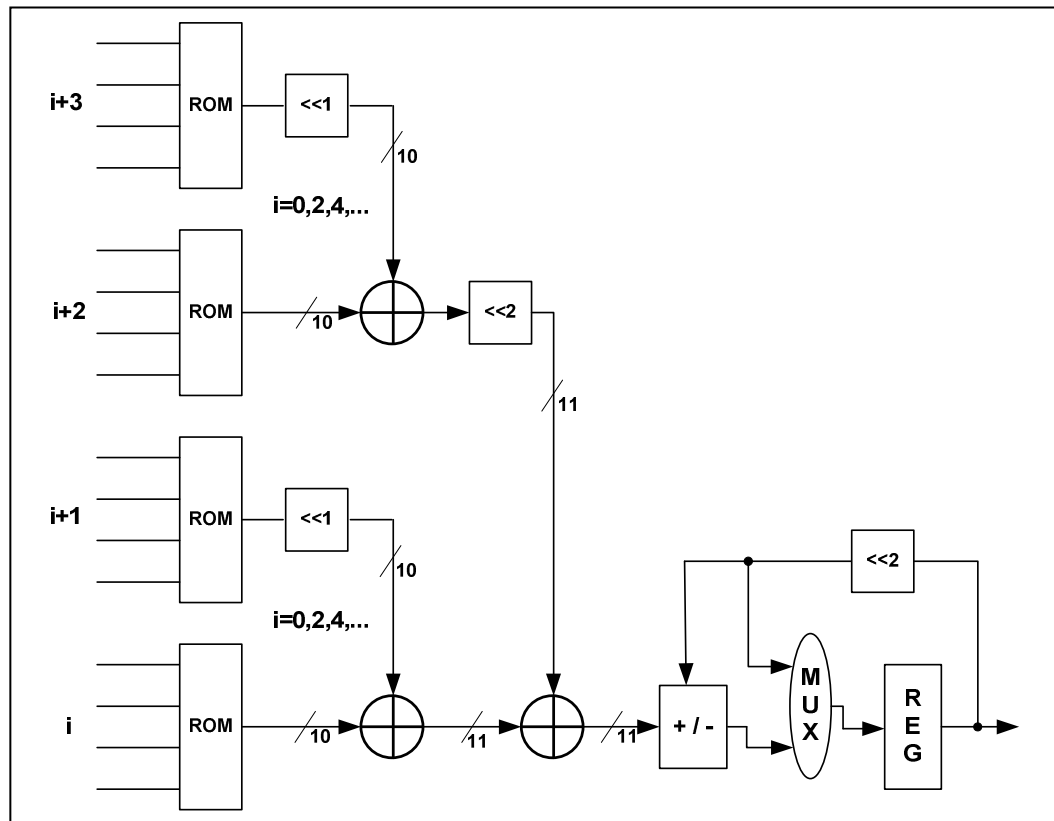

Figure 5.   Four ROM RAC Architecture

The first ROM is used by the even indexed input bits, the second one is used by odd indexed input bits, the third by every other even indexed and the fourth by every other odd indexed input bits. This RAC scheme reduces by half the execution cycles compared to the previous one, at the expense of an increase in hardware resources. This second architecture needs 65 clock cycles in order to process an 8x8 image block.

IV.   SYNTHESIS RESULTS

Initially, a behavioral model was developed that simulated the behavior of the 2D DCT algorithm. This model has been used in order to verify the correct functionality of the proposed architectures of the 2D DCT.

The proposed architectures have been captured using VHDL. The VHDL codes are implemented and synthesized with the Xilinx ISE 13.1 tool. The target FPGA device was XC7VX330T-3FFG1157. The measurements are focused on the design throughput and the consumed FPGA area resources.

Table I presents the synthesis results of the proposed architectures. The proposed architecture that uses the RAC structure with 2 ROMs is shown as 2D_DCT_2ROM, while the proposed architecture that uses the RAC structure with 4 ROMs it was symbolized as 2D_DCT_4ROM.

TABLE I.   SYNTHESIS RESULTS

FPGA Device	XC7VX330T-3FFG1157	
FPGA Resources	**Architectures**	
	2D_DCT_2ROM	2D_DCT_4ROM
# Slice Registers	1354	1110
# Slice LUTs	1786	2021
Freq (MHz)	338.5	256
Bit rate (Samples/sec)	206	252

The 2D_DCT_2ROM architecture consumes less hardware resources (1786 slice LUT) and yields better clock frequency (338.5 MHz) as compared to 2D_DCT_4ROM architecture that consumes 2021 slice LUT and achieves a clock frequency up to 256 MHz. The 2D_DCT_2ROM architecture needs 105 clock cycles to manage 64 samples (an 8x8 image), which corresponds to a bit rate equal to 206 Samples per second. On

the other hand, the second architecture achieves lower clock frequency managing to process more samples (252 samples per second) due to the fact that it needs less clock cycles (65 cycles). This means that the second architecture improves the time performance by 22 % with a penalty in hardware resources of 13 %.

Comparisons with other existing 2D DCT FPGA architectures are given in Table II. Some of those architectures use DA techniques [9]–[10], while others use traditional multipliers [8], [11]. In [12] an ASIC implementation was used.

TABLE II.   COMPARISONS

Architecture	# Slice Registers	# Slice LUTs	Freq (MHz)	Bit rate (Samples/sec)
[8]	1551	1239	102	70
[9]	1443	2398	83	73
[10]	30720	30720	100	63
[11]	NA	NA	150	143
[12]	11.7 kgates (ASIC)		300	112
2D_DCT_2ROM	1354	1786	338.5	206
2D_DCT_4ROM	1110	2021	256	252

The 2D_DCT_2ROM architecture consumes 1786 slice LUT and achieves a clock frequency of up to 338.5 MHz while the 2D_DCT_4ROM architecture consumes 2021 slice LUT and achieves a clock frequency of up to 256 MHz. The 2D_DCT_2ROM architecture needs 105 clock cycles to process an 8x8 image block with a bit rate equal to 206 samples per second, while the 2D_DCT_4ROM achieves a clock frequency of 256 MHz and processes an 8x8 image block in 65 cycles at a bit rate of up to 252 samples per second. The implementation in [8] needs 94 clock cycles for the 2D DCT for a given 8x8 block but uses 8 18x18 multipliers. This results in a major overhead in speed. So, it achieves a frequency of up to 102 MHz and a bit rate of up to 70 samples per second. The work in [9] presents an multiplierless DCT architecture that processes an 8x8 image block in 94 clock cycles resulting in a lower frequency (83 MHz) and thus in a lower bit rate (73 samples per second). The authors in [10] implement both 2D DCT and Motion Estimation, which results in higher hardware resource requirements. With a clock frequency of 100 MHz, the architecture reaches a bit rate of 63 samples per sec. In [11] a Virtex-5 implementation for the 2D DCT that uses 64 multipliers is presented. It takes 67 cycles for an 8x8 block and achieves a rate equal to 143 samples per second. Finally, in [12], an ASIC implementation (@ 0.35-μm CMOS process) for DCT and IDCT is presented. This implementation needs 172 clock cycles for an 8x8 image block at 300 MHz frequency, thus achieving a rate of up to 112 samples per clock.

It is obvious that the proposed architectures outperform the previous FPGA implementations in terms of time performance and achieve similar level of hardware resources. The most important is the fact that both accomplished higher frequency and time performances compared to the ASIC implementation.

V.   CONCLUSIONS

Two high speed FPGA architectures for the 2D DCT computation are presented in this paper. Both architectures use Distributed Arithmetic in order to replace the multipliers that are needed by the DCT algorithm. In the 2D_DCT_2ROM architecture two ROMs are used in order to double the performance, while in the 2D_DCT_4ROM four ROMs are used to quadruple the performance. The synthesis results prove that both architectures are very good choices for applications of high time performance requirements.

REFERENCES

[1] J. Lee and H. Kalva, "Video Coding Techniques and Standards", In Furht B. (Ed.) Encyclopedia of Multimedia, Springer-Verlag Berlin Heidelberg, 2008.

[2] T. Wiegand, G. J. Sullivan, G. Bjontegaard and A. Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, Issue 7, pp. 560-576, July 2003.

[3] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22, No. 12, pp. 1649-1668, December 2012.

[4] S. A. White, "Applications of Distributed Arithmetic to Digital Signal Processing: A Tutorial Review", IEEE ASSP Magazine, Vol. 6, Issue. 3, pp. 4-19, July 1989.

[5] T. Xanthopoulos, and A. P. Chandrakasan, "A Low-Power DCT Core Using Adaptive Bitwidth and Arithmetic Activity Exploiting Signal Correlations and Quantization", IEEE Journal of Solid-State Circuits, Vol. 35, No. 5, pp. 740-750, May 2000.

[6] W. H. Chen, C. H. Smith, and S. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform," IEEE Tranactions on Communications, vol. COM-25, pp. 1004–1009, September 1977.

[7] K. R. Rao and P. Yip, "Discrete Cosine Transform: Algorithms, Advantages, Applications", New York: Academic Press, 1990.

[8] T. Pradeepthi and A. P. Ramesh, "Pipelined Architecture of 2D-DCT, Quantization and Zigzag Process for JPEG Image Compression Using VHDL", International Journal of VLSI Design & Communication Systems, Vol. 2, Issue 3, pp. 99-110, 2011.

[9] A. F. Mahmood and A. M. Salih, "FPGA Implementation of Multiplierless DCT/IDCT Chip", Al-Rafadain Engineering Journal, Vol. 19, Issue 4, pp. 55-67, 2011.

[10] J. Huang, M. Parris, J. Lee and R. F. DeMara, "Scalable FPGA-based Architecture for DCT Computation Using Dynamic Partial Reconfiguration", ACM Transactions on Embedded Computing Systems (TECS), Vol. 9, Issue 1, pp. 1-18, 2009.

[11] A. Fakhari and M. Fathy, "A Two Level Architecture for High Throughput DCT-Processor and Implementing on FPGA", International Conference on ReConFigurable Computing and FPGAs (ReConFig 2010), pp. 115-120, Cancun, Quintana Roo, Mexico, 13-15 Dec. 2010.

[12] G. A. Ruiz, J. A. Michell and A. M. Burón, "High Throughput 2D DCT/IDCT Processor for Video Coding", IEEE International Conference on Image Processing (ICIP 2005), Genoa, Italy, pp. 1036-1039, 11-14 Sept. 2005.