RNS Implementation of Two Dimensional Discrete Cosine Transform over FPL Devices

P. G. FERNÁNDEZ [1], J. RAMÍREZ [2], A. GARCÍA [3], L. PARRILLA [2], A. LLORIS [2]

[1] Dept. of Electrical Engineering
Universidad de Jaén
Escuela Politécnica Superior
23071, Jaén, Spain

[2] Dept. of Electronics and Computer Technology
Universidad de Granada
Campus Universitario Fuentenueva
18071, Granada, Spain

[3] School of Computer Engineering
Universidad Autónoma de Madrid
Ciudad Universitaria de Cantoblanco
28049, Madrid, Spain

Abstract: - The FPL implementation of an 8×8 Two Dimensional Discrete Cosine Transform (2D-DCT) processor based on the Residue Number System (RNS) is presented in this paper. It makes use of a Fast Cosine Transform (FCT) algorithm that requires a single multiplication stage for each signal path, while most other algorithms include paths with more than one multiplication. The row-column decomposition technique is used and each 1D-DCT processor requires only 14 multipliers and 32 adders and subtractors. Performance and area were analysed through synthesis and simulation over Altera FLEX10KE FPL devices. The proposed RNS-based 2D-DCT processor provides a relevant throughput improvement compared to the equivalent 2's complement system implementation when 8-bit moduli are used.

Key-Words: - Residue Number System, Discrete Cosine Transform, Field Programmable Logic.

1 Introduction

The Discrete Cosine Transform (DCT) [1] is one of the most widely used transforms in digital signal processing applications due to its decorrelation and energy compaction properties. Thus, this operation has been adopted by most image and video compression standards, such as JPEG (Joint Photographic Experts Group), ITU-T H.261/H.263 and MPEG (Moving Pictures Experts Group). The most commonly considered transform is the two-dimensional DCT (2D-DCT), used as a standard processing component in many applications. Several hardware-design methods have been developed for the implementation of the 2D-DCT. The most efficient hardware implementation is based on the use of the DCT separability property, with the 2D-DCT being computed by means of several one-dimensional DCTs (1D-DCT). One solution for the computation of the 1D-DCT is the FCT algorithm. Numerous FCT algorithms have been reported, ranging from those that minimize the number of additions and multiplications to those that reduce the number of multiplication stages in the signal path.

RNS is a powerful solution to achieve constant performance while increasing precision [2]. FPL devices provide Digital Signal Processing (DSP) arithmetic support with fast carry chains (Xilinx XC4000, Altera FLEX) that are used to implement Multiply-and-Accumulate (MAC) blocks at high speed [3]. Thus, FPGAs (Field Programable Gate Arrays) are a good solution to implement computationally intensive algorithms, offering the flexibility of a programmable solution and the performance of a custom VLSI chip. In this way, a number of tasks performed in the main DSP computation procedures can be accelerated through the use of efficient RNS arithmetic units improved for the MAC opperation [4, 5], showing improved results over the corresponding binary systems. Indeed, an RNS-based direct implementation of the separable DCT has recently been reported [6], showing a substantial throughput increase when compared to a 2's complement implementation. This

paper shows a fast implementation of the 8×8 pixel 2D-DCT through the use of an FCT algorithm and the RNS over FPL. The architecture presents high regularity and throughput, a reduced dynamic range and is ideal for hardware implementation.

2 RNS background

An RNS [7] is defined by a set of L positive and pairwise relatively prime integers $\{m_1, m_2, \ldots, m_L\}$, called moduli. Let $M = \prod_{i=1}^{L} m_i$, the ring of integers modulo M, $Z(M)$, represents the dynamic range of the RNS. Thus, any positive integer $X \in Z(M)$ is uniquely represented by the L-tuple $[x_1, x_2, \ldots, x_L]$ of its residues, where $x_i = X \bmod m_i$.

Given two integers X, $Y \in Z(M)$, and their corresponding residue representations $[x_1, x_2, \ldots, x_L]$ and $[y_1, y_2, \ldots, y_L]$, arithmetic in the RNS is defined by:

$$X \circ Y \leftrightarrow \left[\left| x_1 \circ y_1 \right|_{m_1}, \left| x_2 \circ y_2 \right|_{m_2}, \ldots, \left| x_L \circ y_L \right|_{m_L} \right] \quad \textbf{(1)}$$

where $\circ$ represents either addition, subtraction or multiplication. In this way, RNS arithmetic is defined over the direct sum of integer rings modulo m_i, $Z(m_i)$, and it is performed in parallel without dependencies between the residue digits.

Conversion from the residue representation $[x_1, x_2, \ldots, x_L]$ to its integer value X is based on the Chinese Remainder Theorem (CRT). This states that there exists a unique $X \in Z(M)$ for the given residue representation and that it can be computed as:

$$\left| X \right|_M = \left| \sum_{j=1}^{L} \hat{m}_j \left| \frac{x_j}{\hat{m}_j} \right|_{m_j} \right|_M \qquad \hat{m}_j \equiv \frac{M}{m_j} \quad \textbf{(2)}$$

Thus, the RNS provides a fundamental methodology for the partitioning of a large dynamic range system into a number of smaller but independent channels over which computations may be performed in parallel.

3 FCT Algoritm for the 1D-DCT

The basic N point 1D-DCT is defined by:

$$X(u) = \sqrt{\frac{2}{N}} e(u) \sum_{i=0}^{N-1} x(i) \cos \frac{u(2i+1)\pi}{2N} \quad \textbf{(3)}$$

$$e(0) = \frac{1}{\sqrt{2}} \qquad e(1) = e(2) = \ldots = e(N-1) = 1$$

When applied to an image block of size $N \times N$, the 2D-DCT is computed as:

$$X(u,v) = \frac{2e(u)e(v)}{N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x(i,j) \cos \frac{u(2i+1)\pi}{2N} \cos \frac{v(2j+1)\pi}{2N} \quad \textbf{(4)}$$

$$u,v = 0, 1, \ldots, N-1$$

where $x(i, j)$ is an $N \times N$ matrix of pixels and $X(u, v)$ is the corresponding transformed matrix.

The 2D-DCT is a separable transform, and the row-column decomposition technique can be used [8]. It is possible to calculate the $N \times N$ 2D-DCT by successively performing N 1D-DCTs on the rows and then, N 1D-DCTs on the resulting columns, as shown in Fig. 1.

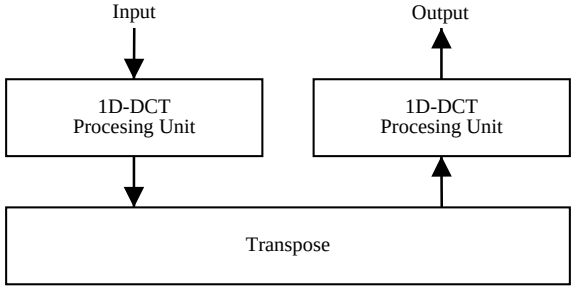

Fig.1. 2D DCT computation through the row-column decomposition technique.

A frequently used block size in image coding is 8×8 since it represents a good compromise between coding efficiency and hardware complexity. Numerous FCT algorithms for the calculation of the DCT have been proposed. The aim of developing these fast algorithms is to reduce the number of arithmetic operations and, consequently, to simplify hardware realizations. In this way, a 2D-DCT of an 8×8 matrix of pixels can be performed with 8 8-pixel 1D-DCTs, followed by a matrix transposition and 8 8-pixel 1D-DCTs of the transposed matrix.

By exploiting the symmetry in the cosine factor matrix and introducing zero terms within the matrix, a reduction in the number of multiplications is obtained, so that the 8-pixel 1D-DCT can be computed as:

$$\begin{bmatrix} X(0) \\ X(4) \\ X(2) \\ X(6) \end{bmatrix} = \begin{bmatrix} c_4 & c_4 & 0 & 0 \\ c_4 & -c_4 & 0 & 0 \\ 0 & 0 & c_6 & c_2 \\ 0 & 0 & -c_2 & -c_6 \end{bmatrix} \cdot \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix}$$

$$\begin{bmatrix} X(7) \\ X(5) \\ -X(1) \\ X(3) \end{bmatrix} = \begin{bmatrix} c_7 & c_5 & -c_1 & c_3 \\ c_5 & c_1 & c_3 & c_7 \\ -c_1 & c_3 & -c_7 & -c_5 \\ c_3 & c_7 & -c_5 & -c_1 \end{bmatrix} \cdot \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \end{bmatrix} \quad \textbf{(5)}$$

where $x(i)$ are the pixel data, $X(u)$ are the transformed coefficients, $c_k = \frac{1}{2}\cos\left(k \cdot \frac{\pi}{16}\right)$ and:

$$\begin{aligned} h_1 &= x(0)+x(7)+x(3)+x(4) & i_1 &= x(0)-x(7) \\ h_2 &= x(1)+x(6)+x(2)+x(5) & i_2 &= x(6)-x(1) \\ h_3 &= x(1)+x(6)-x(2)-x(5) & i_3 &= x(3)-x(4) \\ h_4 &= x(0)+x(7)-x(3)-x(4) & i_4 &= x(2)-x(5) \end{aligned} \quad \textbf{(6)}$$

Using the standard rotator product [9, 10], the number of multiplications is reduced. Thus, 4 multiplications and 2 additions can be replaced by 3 multiplications and 3 additions.

Consequently, the matrix vector product technique given in equation (5) leads to an efficient FCT algorithm that requires only 14 multiplications and 32 adders and substractors, as shown in following equations:

$$\begin{bmatrix} X(0) \\ X(4) \\ X(2) \\ X(6) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} k_1 y_1 \\ k_2 y_2 \\ k_3 h_3 \\ k_4 y_3 \\ k_5 h_4 \end{bmatrix}$$

$$\begin{bmatrix} X(7) \\ X(5) \\ -X(1) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} k_6 i_1 \\ k_7 z_1 \\ k_8 i_2 \\ k_9 z_2 \\ k_{10} z_3 \\ k_{11} z_4 \\ k_{12} i_3 \\ k_{13} z_5 \\ k_{14} i_4 \end{bmatrix} \quad \textbf{(7)}$$

where:

$$\begin{aligned} y_1 &= h_1 + h_2 & y_2 &= h_1 - h_2 \\ y_3 &= h_3 + h_4 & z_1 &= i_1 + i_2 \\ z_2 &= i_1 + i_3 & z_3 &= i_1 + i_2 + i_3 + i_4 \\ z_4 &= i_2 + i_4 & z_5 &= i_3 + i_4 \\ k_1 &= c_4 & k_2 &= c_4 \\ k_3 &= c_6 - c_2 & k_4 &= c_2 \\ k_5 &= c_2 + c_6 & k_6 &= c_1 + c_3 - c_5 + c_7 \\ k_7 &= -c_3 + c_5 & k_8 &= c_1 + c_3 - c_5 - c_7 \\ k_9 &= -c_1 - c_3 & k_{10} &= c_3 \\ k_{11} &= -c_3 + c_7 & k_{12} &= c_1 + c_3 + c_5 - c_7 \\ k_{13} &= -c_3 - c_5 & k_{14} &= -c_1 + c_3 + c_5 - c_7 \end{aligned} \quad \textbf{(8)}$$

Moreover, an attractive feature of this algorithm is that each computation path contains only one multiplication stage, and these multiplications can be executed in parallel, as shown in Figure 2.

4 RNS-Based 2D-DCT

An 8×8 pixel 2D-DCT processor has been implemented using the RNS on FPL devices. The 2D-DCT is designed using the row-column decomposition method. It consists of two 1D-DCT processors based on the FCT algorithm described above, incorporating an intermediate transposition unit, which allows the necessary transposition to be carried out on the data before they are fed to the second 1D-DCT unit. The implementation of the transposition unit consists of an 8×8 matrix of registers and multiplexers interconnected to allow the transposition of parallel input data [11]. The multiplexers select either the upper or lower paths during eight consecutive clock cycles to enable data transposition.

Using the alternative FCT algorithm described above, the modulo m_j 8-pixel RNS 1D-DCT can be represented by the architecture shown in Fig. 3. Under this scheme, each signal path requires just a single multiplication stage, while most other algorithms have paths with more than one multiplication [12, 13]. Thus, the dynamic range and, consequently, the number of RNS channels required to enable it, is reduced. This 1D-DCT architecture requires only 14 multipliers and 32 adders or subtractors, less than other parallel designs [14, 15] that require 16 and 24 multiplications. The cost of having no more than one multiplication per path is, therefore, three additional multiplications and additions, compared to the currently published algorithms that include signal paths with two multiplications. Moreover, by scaling the DCT matrix given in equation (5) by c_4 [10], the complexity of the algorithm can be reduced. Then, 2 multiplications can be eliminated, thus reducing the number of fixed coefficient multipliers to only 12. Linear combinations of the DCT coefficients are precalculated and stored in ROMs, so the DCT can be calculated with Look-Up Table (LUT) multipliers. The structure is highly regular and contains only registers, adders, substractors and memories.

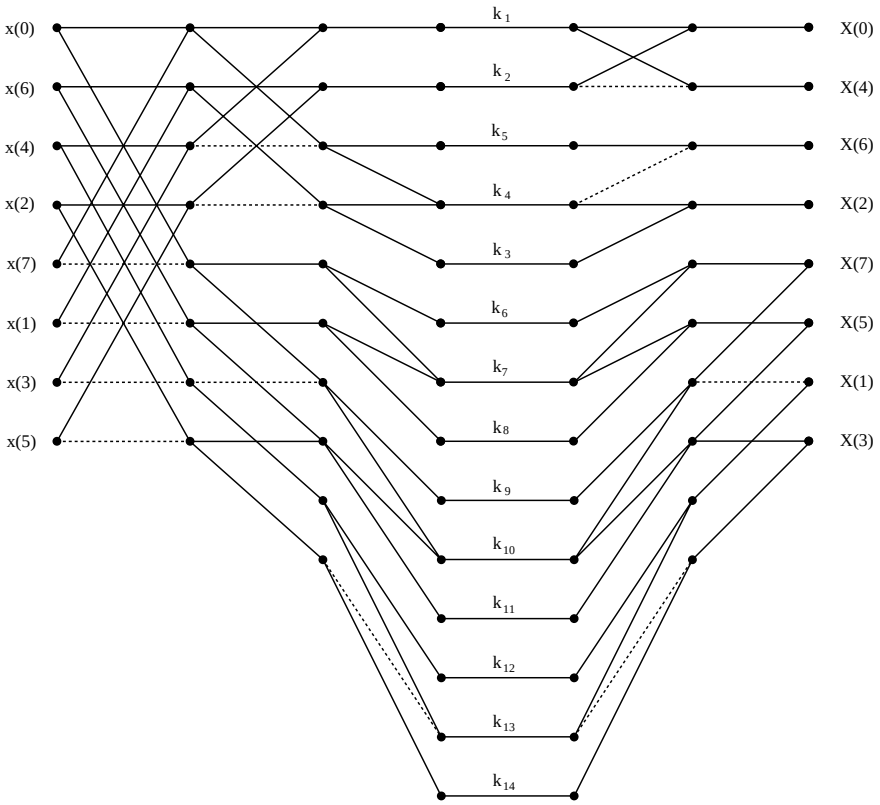

Fig.2. Flow graph for fast implementation of 1D-DCT**.**

5 2D-DCT FPL Implementation

RNS and binary 2's complement FPL-based implementations of the proposed 8×8 pixel 2D-DCT processor using Altera FLEX10KE [3] devices have been developed using structural-level VHDL synthesis to compare parameters such as area and performance. These devices contain an embedded array to implement memory and specialized logic functions, and a logic array to implement general logic. The embedded array consists of a series of EABs (Embedded Array Blocks). When an EAB is used as a memory it provides 4096 bits. The logic array consists of LEs (Logic Elements), with each LE providing a 4-input LUT, a programmable flip-flop and a dedicated signal path for carry and cascade chains. When an RNS consisting of 8-bit moduli is used, it is possible to store a $2^8 \times 8$ fixed cosine coefficient multiplication in a single EAB. Moreover, modular adders do not use memory resources and take full advantage of short carry chains. At the same time, flip-flops included in the LEs mean that additional resources are not required for pipelining.

Using 7-bit cosine coefficients and 8-bit signal samples, the 32−bit dynamic range required by the 2D-DCT processor is supported by the modulus set {256, 255, 253, 251}. Modulus 256 offers hardware reduction in the binary-to-RNS conversion and a considerable memory reduction in the processing channel, since *n*-bit power-of-two modular adders and multipliers are efficiently implemented with conventional *n*-bit binary adders and multipliers, respectively. In this way, the proposed RNS-based 8×8 pixel 2D-DCT processor consists of four 8-bit RNS channels all operating in parallel for maximum performance. The transposition unit, when mapped on FPL devices, takes advantage of the device's architecture by synthesizing the multiplexer logic and the register for each bit in the same LE. Detailed information about the LE and EAB usage for both RNS and binary 2's complement 8×8-pixel 2D-DCT processors and their throughput for several speed grade devices is shown in Table 1. The throughput is measured in millions DCTs per second.

Thus, the proposed RNS-based 2D-DCT processor provided a throughput advantage of up to 147% over the 2's complement binary implementation when two-stage binary multipliers are considered for this option.

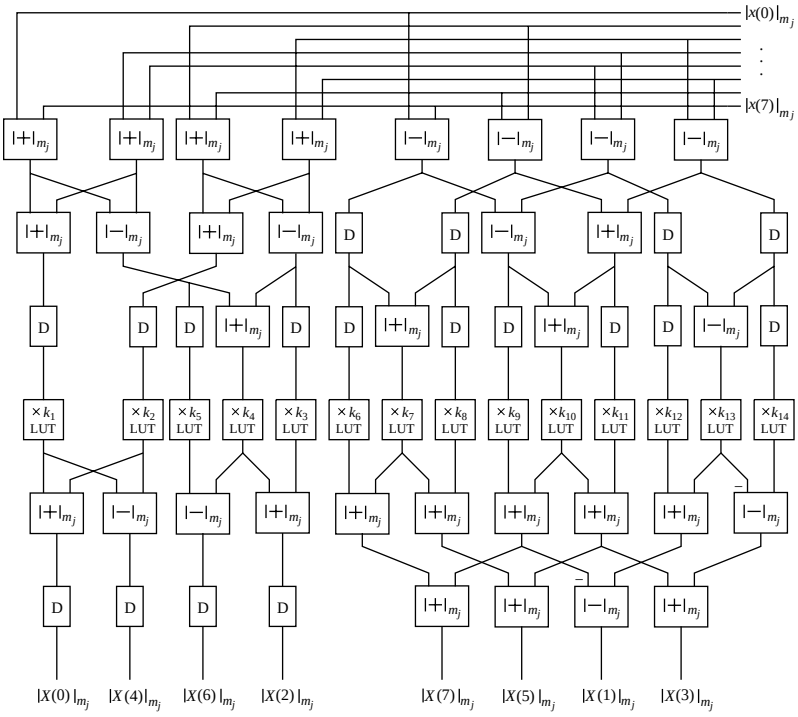

Fig. 3. Modulo m_j channel of the 1D-DCT.

On the other hand, the practical implementation of RNS-based systems has a serious limitation in the conversion stage. Different solutions have been addressed to overcome the conversion drawback. Binary-to-RNS is a fast operation since the input and the selected moduli are 8-bit width. However, RNS-to-binary conversion implies the use of 32-bit word-length modular adders and large multipliers, and a direct implementation of the Chinese Remainder Theorem (CRT) results in excessive hardware usage and degrades the system performance. Thus, the auto-scaling RNS-to-binary converter (ε-CRT) proposed by Griffin *et al* [16] allows these drawbacks to be overcome using only LUTs and conventional binary adders.

7 Conclusion

The synergy between RNS and FPL devices has been a subject of study recently. This paper shows a novel, regular and compact 8×8 2D-DCT RNS processor that benefits from the row-column decomposition technique, along with a high-throughput RNS 1D-DCT architecture based on an improved FCT algorithm with only one multiplier stage in the signal path. Thus, the dynamic range required is reduced and, consequently, the number of parallel modulo 2D-DCT processors. Each 1D-DCT requires only 14 multipliers and 32 adders and subtractors, less than other parallel designs. When FPL devices are considered to map this architecture, a transposition unit based on an 8×8 matrix of registers and multiplexers can be used.

Parameters such as area and performance were assessed by synthesizing both binary 2's complement and RNS 2D-DCT processors using VHDL and Altera FLEX10KE devices. Thus, the proposed RNS-based 2D-DCT processor provided a throughput advantage of up to 147%. This advantage is due to the efficient use of the device's built-in tables for the mapping of fixed coefficient products for 8-bit moduli. Finally, RNS-to-binary conversion can be carried out through an auto-scaling ε-CRT converter without degrading the performance of the proposed architecture.

Acknowledgement

The authors were supported by the Comisión Interministerial de Ciencia y Tecnología (SPAIN) under project PB98-1354. CAD tools and supporting material were provided by Altera Corp. under the Altera University Program.

	8-bit RNS channel architecture			Modulo 256 RNS channel			Pipelined binary architecture (Two stage)		
	LEs	EABs	Throughput	LEs	EABs	Throughput	LEs	EABs	Throughput
Grade -1	2355	14×2	128.20	2354	-	131.58	5270	-	51.94
Grade -2	2355	14×2	98.03	2354	-	102.04	5270	-	45.55
Grade -3	2355	14×2	69.93	2354	-	75.18	5270	-	36.36

Table 1. 2D-DCT RNS-based implementation over Altera FLEX10KE FPL devices. Performance and LE and EAB usage.

References:

[1] N. Ahmed, T. Natarajan and K. R. Rao. "Discrete cosine transform". *IEEE Trans. Comput.*, vol. C-23, pp 90-94, Jan. 1974.

[2] M. Soderstrand, W. Jenkins, G. A. Jullien, F. J. Taylor, *Residue Number System Arithmetic: Modern Applications in Digital Signal Processing*. IEEE Press, 1986.

[3] Altera Corporation, *1998 Data Book*, 1998.

[4] J. Ramírez, A. García, P. G. Fernández, L. Parrilla and A. Lloris, "A New Architecture to Compute the Discrete Cosine Transform Using the Quadratic Residue Number System", in *International Symposium on Circuits and Systems*, Geneva, Switzerland, May 2000.

[5] J. Ramírez, A. García, P. G. Fernández, L. Parrilla, A. Lloris, "RNS-FPL merged architectures for orthogonal DWT", *Electron. Lett.*, Vol. 36, No. 14, pp. 1198-1199, 2000.

[6] P. G. Fernández, A. García, J. Ramírez, L. Parrilla and A. Lloris, "A RNS-based Matrix-Vector-Multiply FCT Architecture for DCT Computation", in *43rd Midwest Symposium on Circuit and System* MWSCAS 2000.

[7] N. S. Szabo, R. I. Tanaka, *Residue Arithmetic and Its applications to Computer Technology*. McGraw-Hill, New York, 1967.

[8] K. R. Rao and P. Yip. *Discrete cosine transform: Algorithms, advantages, applications*. Academic Press,Inc., 1990.

[9] P. Thitimajshima, *Implementation of Two Dimensional Discrete Cosine Transform Using Field Programmable Gate Array*. Signal Processing, Communications and Computer Science, World Scientific and Engineering Society Press, págs. 47-50. 2000.

[10] C. Loeffler, A. Ligtenberg, and G. S. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications", in *Proc. of 1989 IEEE International Conference on Acoustics, Speech and Signal Processing Systems*, pp 988-991.

[11] P. G. Fernández, A. García, J. Ramírez, L. Parrilla and A. Lloris, **"**A New RNS Architecture for the Computation of the Scaled 2D-DCT on Field-Programmable Logic**",** in *34rd Asilomar Conf. on Signals, Systems and Computers*, 2000.

[12] B. G. Lee, "A New Algorithm to Compute the Discrete Cosine Transform" , *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 32, no. 6, pp 1243-1245, Dec. 1984.

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

[14] P. C. Jain, W. Schlenk, and M. Riegel, "VLSI implementation of two-dimensional DCT processor in real time for video codec" , *IEEE Trans. on Consumer electronics*, vol. 38, no. 3, pp 537-540, Aug. 1992.

[15] C. Y. Hung and P. Landman, "Compact Inverse discrete cosine transform circuit for MPEG video decoding", in *Proc. of 1997 IEEE Signal Processing Systems: Design and Implementation*, Leicester, U.K., pp 364-373, Nov. 1997.

[16] M. Griffin, F. J. Taylor and M. Sousa, "New scaling algorithms for the Chinese Remainder Theorem", *Proc of 22nd Asilomar Conference on Signals, Systems and Computers* (Pacific Grove, CA), Oct. 1988.Texas Instruments, *TMS320C5x Users's Guide*, 1993.