Adv. Radio Sci., 5, 305–311, 2007
www.adv-radio-sci.net/5/305/2007/

Advances in
Radio Science

A low-power and high-quality implementation of the discrete cosine transformation

B. Heyne and J. Götze

University of Dortmund, Information Processing Lab, Otto-Hahn-Str. 4, 44227 Dortmund, Germany

Abstract. In this paper a computationally efficient and high-quality preserving DCT architecture is presented. It is obtained by optimizing the Loeffler DCT based on the Cordic algorithm. The computational complexity is reduced from 11 multiply and 29 add operations (Loeffler DCT) to 38 add and 16 shift operations (which is similar to the complexity of the binDCT). The experimental results show that the proposed DCT algorithm not only reduces the computational complexity significantly, but also retains the good transformation quality of the Loeffler DCT. Therefore, the proposed Cordic based Loeffler DCT is especially suited for low-power and high-quality CODECs in battery-based systems.

1 Introduction

Recently, many kinds of digital image processing and video compression techniques have been proposed in the literature, such as JPEG, Digital Watermark, MPEG and H.263 (Conzalez and Woods, 2001; Richardson, 2002). All the above standards require the Discrete Cosine Transform (DCT) Conzalez and Woods (2001) to aid image/video compression. Therefore, the DCT has become more and more important in today's image/video processing designs.

In the past few years, much research has been done on low power DCT designs (Li and Lu, 1996; Hsiao et al., 2005; August and Ha, 2004; Jeong et al., 2004; Shams et al., 2002; Fanucci and Saponara, 2002). One of the most popular ways to realize the fast DCT (FDCT) is to use the Flow-Graph Algorithm (FGA) for VLSI-implementation (Chen et al., 1977; Wang, 1984). Loeffler et al. (1989) has proposed a low-complexity FDCT/IDCT algorithm based on FGA that requires only 11 multiply and 29 add operations. However, the multiplications consume about 40% of the power and almost 45% of the total area (Sung et al., 2005). In this regard, Tran (2000) proposed the binDCT which approximates multiplications with add and shift operations. It only consumes about 38% of the power of the Loeffler DCT. However, it also looses about 3 dB in PSNR compared to the Loeffler DCT (Sung et al., 2005).

Jeong et al. (2004) has proposed a Cordic based implementation of the DCT. COordinate Rotation DIgital Computer (Cordic) is an algorithm that can be used for the evaluation of various functions in signal processing (Volder, 1959; Walther, 1971). In addition, the Cordic algorithm is highly suited for VLSI-implementation.

In this paper we propose a computationally efficient and high-quality Cordic based Loeffler DCT architecture, which is optimized by taking advantage of certain properties of the Cordic algorithm and its implementations (Goetze and Hekstra, 1995). It only requires 38 add and 16 shift operations. The resulting DCT algorithm not only reduces computational complexity significantly, but also retains the good transformation quality of the Loeffler DCT. Therefore, the presented Cordic based Loeffler DCT implementation is especially suited for low-power and high-quality CODECs.

This paper is organized as follows. Section 2 briefly introduces the algorithms of the DCT, Loeffler DCT and Cordic based DCT. In Sect. 3, we will present the proposed Cordic based Loeffer DCT algorithm. The experimental results are shown in Sect. 4, while Sect. 5 concludes this paper.

Correspondence to: B. Heyne
(benjamin.heyne@uni-dortmund.de)

Published by Copernicus Publications on behalf of the URSI Landesausschuss in der Bundesrepublik Deutschland e.V.

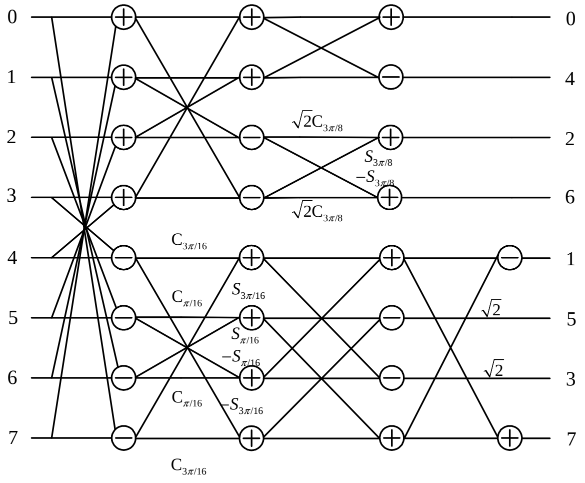

Fig. 1. Flow graph of an 8-point Loeffler DCT.

2 DCT algorithms

2.1 The DCT background

The two dimensional DCT in Eq. (1) transforms an 8 × 8 block sample from spatial domain $f(x, y)$ into frequency domain $F(k, l)$.

$$F(k,l) = \frac{1}{4}C(k)C(l)\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)$$

$$\cdot \cos[\frac{(2x+1)k\pi}{16}] \cos[\frac{(2y+1)l\pi}{16}] \quad (1)$$

$$C(m) \quad = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } m = 0 \\ 1 & \text{otherwise.} \end{cases}$$

Since computing the above 2-D DCT by using matrix multiplication requires 8^4 multiplications, a commonly used approach in hardware designs to reduce the computational complexity is row-column decomposition. The decomposition performs row-wise one-dimensional (1-D) transform followed by column-wise 1-D transform with intermediate transposition. An 8-point 1-D DCT can be expressed as follows:

$$F(k) = \frac{1}{2}C(k)\sum_{x=0}^{7} f(x)\cos[\frac{(2x+1)k\pi}{16}]$$

$$C(k) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if k = 0} \\ 1 & \text{otherwise.} \end{cases} \quad (2)$$

This decomposition approach has two advantages. Firstly the number of operations is significantly reduced. Secondly, with regard to the implementation, the original 1-D DCT can be replaced easier by more efficient DCT algorithms.

2.2 MAC based Loeffler DCT

Many 1-D flow graph algorithms have been reported in the literature (Chen et al., 1977; Wang, 1984). The Loeffer 1-D 8-point DCT algorithm (Loeffler et al., 1989) requires 11 multiplications and 29 additions as shown in Table 3. The flow graph of the Loeffler DCT is illustrated in Fig. 1, with $C_x = \cos(x)$ and $S_x = \sin(x)$. One of its variations is adopted by the Independent JPEG Group (JPE, 1998) for their implementation of the popular JPEG image coding standard. Note that this factorization requires a uniform scaling factor of $\frac{1}{2\sqrt{2}}$ at the end of the flow graph to obtain the original DCT coefficients. In the 2-D transform this scaling factor becomes $\frac{1}{8}$ which can be easily implemented by a shift operation. Although the Loeffler DCT requires multipliers, which will result in larger power dissipation and area, it offers better rate distortion than the other approaches. Therefore it is especially useful for high-quality CODECs.

2.3 Cordic based DCT

The Loeffler DCT achieves good quality transformation results, but on the other hand it needs multiplications which are computationally intensive in both software and hardware implementation. In this regard, one of the popular ways to implement a fast multiplierless approximation of the DCT is using the Cordic algorithm (Jeong et al., 2004; Hsiao et al., 2005). The Cordic has a very regular structure suitable for VLSI design. Figure 2 shows the flow graph of an 8-point Cordic based DCT using six Cordics (Jeong et al., 2004), requiring 104 additions and 84 shift operations as shown in Table 3.

In order to realize a vector rotation for the Cordic algorithm, that is rotating a vector (x, y) by an angle θ, the circular rotation angle is described as

$$\theta = \sum_i \sigma_i \cdot \tan^{-1}(2^{-i}) \quad (3)$$

$$\text{with } \sigma_i = 1, -1.$$

Then, the vector rotation can be performed using the iterative equation given in (Mariatos et al., 1994; Jeong et al., 2004):

$$\begin{aligned} x_{i+1} &= x_i - \sigma_i \cdot y_i \cdot 2^{-i} \\ y_{i+1} &= y_i + \sigma_i \cdot x_i \cdot 2^{-i}. \end{aligned} \quad (4)$$

In Eq. (4), only shift and add operations are required in digital hardware. Next, the results of the rotation iterations need to be compensated (scaled) by a compensation factor s. This is also done using an iterative approach:

$$\begin{aligned} x_{i+1} &= x_i(1 + \gamma_i \cdot F_i) \\ y_{i+1} &= y_i(1 + \gamma_i \cdot F_i) \end{aligned}$$

$$\text{with } \prod_i (1 + \gamma_i \cdot F_i) \cong s \quad (5)$$

$$\text{and } \gamma_i = (0, 1, -1),\ F_i = 2^{-i}.$$

Table 1. Parameters for the DCT based on Jeong et al. (2004).

Angle	$\frac{\pi}{4}$	$\frac{3\pi}{8}$	$\frac{7\pi}{16}$	$\frac{3\pi}{16}$
Rotation iterations $[\sigma_i, i]$ according to Eq. (4)				
1	$-1, 0$	$-1, 2$	$+1, 0$	$-1, 1$
2	–	$-1, 3$	$+1, 1$	$-1, 3$
3	–	$-1, 6$	$+1, 3$	$-1, 10$
4	–	$-1, 7$	$+1, 10$	$-1, 14$
Compensation iterations $[1 + \gamma_i \cdot F_i]$ according to Eq. (5)				
1	$1-\frac{1}{4}$	$1+\frac{1}{32}$	$\frac{1}{2}+\frac{1}{8}$	$1-\frac{1}{8}$
2	$1-\frac{1}{16}$	$1+\frac{1}{128}$	$1+\frac{1}{256}$	$1+\frac{1}{64}$
3	$1+\frac{1}{256}$	$1+\frac{1}{1024}$	$1+\frac{1}{4096}$	$1+\frac{1}{1024}$
4	$1+\frac{1}{512}$	$1+\frac{1}{4096}$	–	$1+\frac{1}{4096}$
5	$1+\frac{1}{4096}$	–	–	–

When using the Cordic to replace the multiplications of the 8-point DCT the angles θ_x are fixed. Therefore, we can skip some unnecessary iterations without losing accuracy. Table 1 shows the detailed number of iterations and compensations for the Cordic based algorithm (Jeong et al., 2004). Although the Cordic based DCT can reduce the number of computations in image/video compression, it still needs more operations than the binDCT does (Tran, 2000).

3 Cordic based Loeffler DCT

Based on our previous work about Cordic based FFTs (Heyne and Goetze, 2004; Heyne et al., 2004), we now propose an optimized Cordic based Loeffler DCT in this paper. This implementation requires only 38 add and 16 shift operations. We have taken the original Loeffler DCT as the starting point for our optimization, because the theoretical lower bound of the number of multiplications required for the 1-D 8-point DCT had been proven to be 11 (Duhamel and H'Mida, 1987).

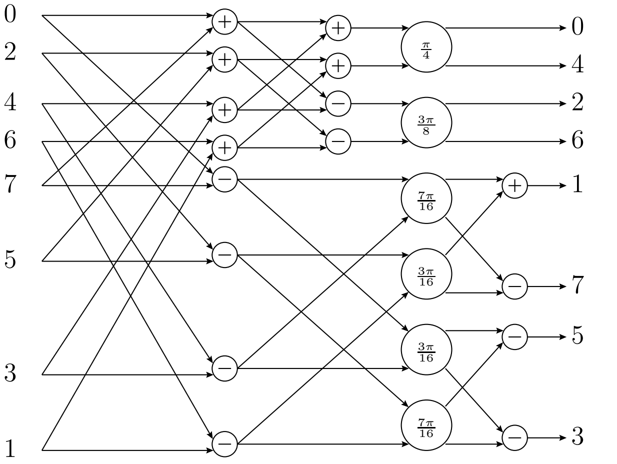

Fig. 2. Flow graph of an 8-point Cordic based DCT (Jeong et al., 2004).

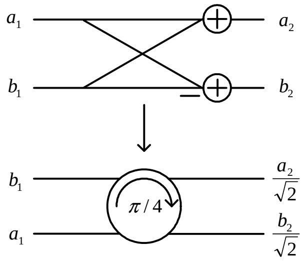

Fig. 3. Cordic rotation of the $\pi/4$ angle.

In order to derive the proposed algorithm, we first consider the butterfly at the beginning of Loeffler's flow graph as shown in Fig. 1. In this case the butterfly can be expressed as:

$$\begin{bmatrix} a_2 \\ b_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ a_1 \end{bmatrix} \tag{6}$$

The first matrix can then be decomposed to

$$\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} = \sqrt{2} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \tag{7}$$

which equals a Cordic rotating the input values by $\pi/4$, followed by a scaling of $\sqrt{2}$ as shown in Fig. 3.

The scaled butterflies with scaling factors $3\pi/8$, $1\pi/16$ and $3\pi/16$ can also be replaced by Cordics using $\theta{=}3\pi/8$, $1\pi/16$ and $3\pi/16$ respectively. Hence, we can replace all butterflies in the Loeffler DCT to derive the pure Cordic based Loeffler DCT as shown in Fig. 4.

The most commonly used DCT-based CODECs for signal processing are usually followed by a quantizer. In this regard we can skip some Cordic iterations without losing visual quality, and shift the compensation steps to the quantization table without using additional hardware.

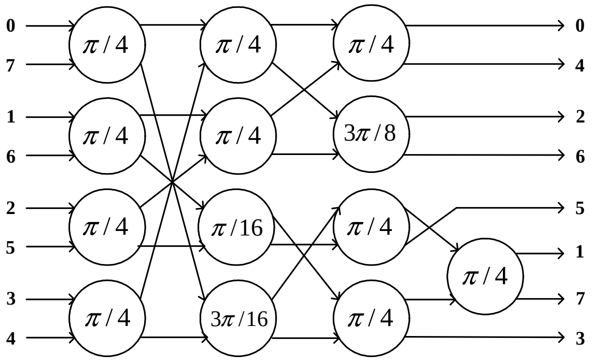

Fig. 4. Pure Cordic based Loeffler DCT architecture.

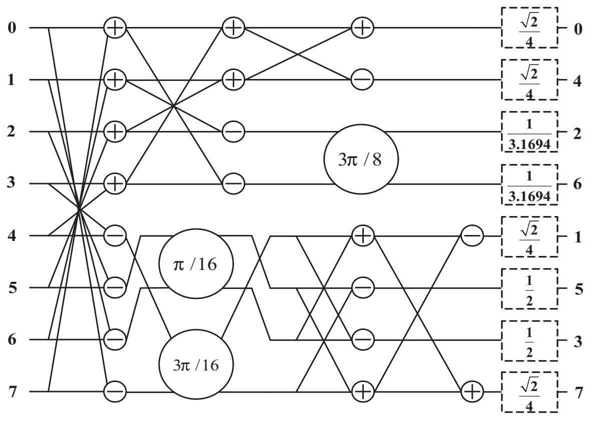

Fig. 5. Flow graph of an 8-point Cordic based Loeffler DCT architecture.

Next, we will start to optimize each rotation angle and reduce the computational complexity.

At first,due to the special structure of the Loeffler DCT, the scaling of $\sqrt{2}$ and the five needed compensation steps as shown in Table 1 can be performed at the end of the flow graph for the angle $\theta=\pi/4$. In other words, the $\pi/4$ rotation only needs two add operations to carry out the Cordic rotation.

Secondly, for the angle $\theta=3\pi/8$ we reduce the number of iterations to three and also shift all compensation steps to the quantizer. Although the optimized $3\pi/8$ rotation will decrease the quality of the results, the influences are not noticeable in video sequence streams or image compression.

Thirdly, when we take a close look at the angle $\theta=\pi/16$, it can be easily observed that the needed compensation of the $\pi/16$ rotation is very close to one. Thus, we can ignore the compensation steps of the $\pi/16$ rotation. Therefore, it only needs two iterations in the Cordic calculation. Unfortunately we can not shift any compensation steps of the $3\pi/16$ rotation to the end of the graph, due to the data correlation between the following stages of the $\pi/16$ and $3\pi/16$ rotations. However, we still can ignore some unnoticeable iterations and compensation steps to reduce the computational complexity of the angle $\theta=3\pi/16$.

Table 2. Cordic based Loeffler DCT – Cordic parameters.

Angle	$\frac{\pi}{4}$	$\frac{3\pi}{8}$	$\frac{\pi}{16}$	$\frac{3\pi}{16}$
Rotation iterations $[\sigma_i, i]$ according to Eq. (4)				
1	$-1, 0$	$-1, 0$	$-1, 3$	$-1, 1$
2	–	$-1, 1$	$-1, 4$	$-1, 3$
3	–	$+1, 4$	-	-
Compensation iterations $[1+\gamma_i \cdot F_i]$ according to Eq. (5)				
1	–	–	–	$1-\frac{1}{8}$
2	–	–	–	$1+\frac{1}{64}$

Table 2 shows the summary of Cordic iterations and compensation steps for the proposed Cordic based Loeffler DCT. In Fig. 5 the optimized flow graph is shown, including the scaling factors incorporated into the quantization table.

It only requires 38 add and 16 shift operations to realize the DCT transformation. In short, we try to ignore some unnoticeable iterations and shift the compensation steps of each angle to the quantizer to derive the optimized Cordic based Loeffler DCT. Moreover, the proposed DCT algorithm not only reduces the computational complexity significantly, but also keeps the high transformation quality as well as the original Loeffler DCT does.

Therefore, the proposed DCT algorithm has the same computational complexity as the binDCT, but as shown in the next Section it can perform as well as the Loeffler DCT in quality.

4 Experimental results

In our experiments we have used different criteria to evaluate four architectures: Loeffler DCT, Cordic based DCT, binDCT-C5 and Cordic based Loeffler DCT. Table 3 summarizes the number of operations of each DCT architecture. It can be easily observed from Table 3 that the proposed DCT reduces the computational complexity significantly compared to the original Loeffler DCT and Cordic based DCT.

Table 3. Complexity of different DCT architectures.

Operation DCT type	Mult	Add	Shift
Loeffler	11	29	0
Cordic [Jeong et al. (2004)]	0	104	82
Cordic Loeffler	0	38	16
binDCT-C5 [Tran (1999)]	0	36	17

In order to demonstrate the quality features of the proposed DCT algorithm, we have also applied it to the video coding standard MPEG 4, by using a publicly available XVID CODEC software (XIV, 2005). The DCT in the CODEC of the selected XVID implementation is based on Loeffler's factorization with floating-point multiplications. In this part we have applied each DCT algorithm to the XVID software, and simulated with some well-known video sequences to show the ability of the proposed approach. Figure 6 shows the average PSNR results of four DCT algorithms from low to high compression ratio (i.e. quantization steps from 1 to 10) with the "Foreman" video sequence.

It can be seen from the PSNR simulation results, that the proposed algorithm performs as well as the Loeffler DCT does. Also, the average PSNR is about 2 dB higher than that of the binDCT-C5. In summary, the overall performance of the Cordic based Loeffler DCT is very similar to the Loeffler DCT in terms of video quality. However, it only requires the same computational complexity as the binDCT-C5 does.

To analyze the performance and the energy consumption of the proposed Cordic based Loeffler DCT, we have modeled the four different DCT architectures as RTL. After synthesizing with Synopsys Design Compiler, we have used Synopsys PrimePower to estimate the power consumption at gate-level.

The simulation results are shown in Table 4. Some important points can be observed easily. Firstly, the proposed DCT architecture only consumed 19% of the area and about 16% of the power of the original Loeffler DCT. Secondly, the proposed DCT architecture occupied the same area as the binDCT-C5. However, it has only about 59% of the power dissipation and half the delay time of the binDCT-C5.

To further display the low-power features of the proposed algorithm, we have analyzed the Power, Power-Delay Product (PDP), Energy-Delay Product (EDP) and Energy-Delay-Delay Product (EDDP). The PDP is the average energy consumed per DCT transformation. A lower PDP means that the power consumption is better translated into speed of each operation and the EDP represents that one can trade increased delay for lower energy of each operation. Finally, the EDDP represents whether or not it results in a voltage-invariant efficiency metric.

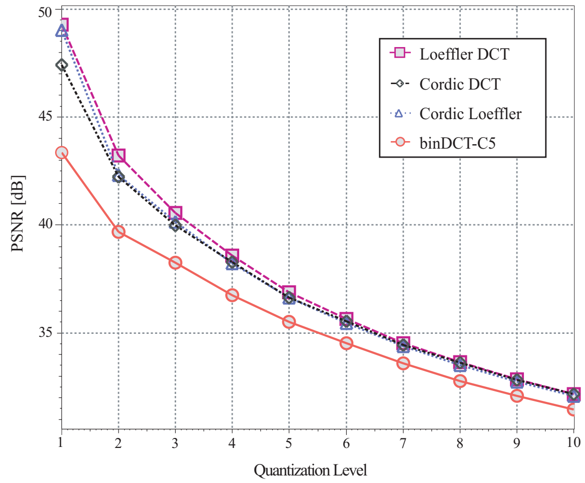

Fig. 6. The average PSNR of the "Foreman" sequence from low to high compression ratio (XVID CODEC).

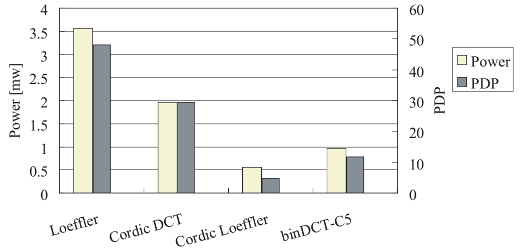

Fig. 7. Experimental results of the Power and PDP.

Figure 7 illustrates the experimental results of the power consumption and the PDP. As illustrated for the PDP the proposed DCT algorithm only consumes about 10% of the PDP of the Loeffler DCT and 16% of the PDP of the Cordic based DCT respectively. It also reduces the PDP by about 59% compared to the binDCT-C5. Figure 8 shows the EDP and the EDDP. As shown, the performance of the proposed DCT algorithm is far superior to the other DCT algorithms, especially in the EDDP.

Hence, the proposed DCT not only reduces the computational complexity significantly, but also achieves the best performance in all criteria.

5 Conclusions

In this paper a low complexity and high quality DCT transformation based on the Cordic algorithm is presented. The

Table 4. Power, Area and Time delay simulation results at gate-level.

DCT Arch. Measures	Loeffler	Cordic based (Jeong et al., 2004)	Cordic based Loeffler	binDCT-C5 (Tran, 1999)
Power (mW)	3.557	1.954	0.5616	0.9604
Area ($GateCount$)	15.06 K	6.66 K	2.81 K	2.83 K
Delay (ns)	13.49	15.08	8.37	12.17

TSMC 0.13-μm at 1.2 V without pipelining.

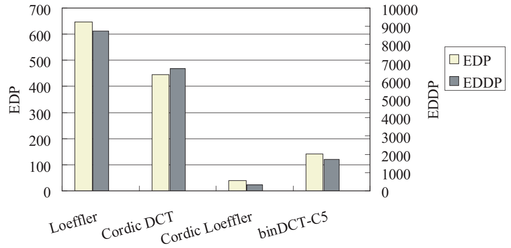

Fig. 8. Experimental results of the EDP and EDDP.

proposed Cordic based Loeffler DCT architecture only requires 38 add and 16 shift operations to carry out the DCT transformation, which is about the same complexity as the binDCT-C5's. The proposed algorithm not only reduces the computational complexity significantly compared to the original Loeffler DCT, it also keeps the good quality transformation result. In this regard, the proposed DCT algorithm is very suitable for low-power and high quality CODECs.

References

The JPEG-6b Website, http://www.ijg.org/, 1998.

The XVID Website, http://www.xvid.org/, 2005.

August, N. J. and Ha, D. S.: Low Power Design of DCT and IDCT for Low Bit Rate Video Codecs, IEEE Transactions on Multimedia, 6, 414–422, 2004.

Chen, W. H., Smith, C., and Fralick, S.: A Fast Computational Algorithm for the Discrete Cosine Transform, 25, 1004–1009, 1977.

Conzalez, R. C. and Woods, R. E.: Digial Image Processing, Prentice-Hall, Inc., New Jersey, 2001.

Duhamel, P. and H'Mida, H.: New 2^n DCT Algorithms Suitable for VLSI Implementation, in: IEEE International Conference on ICASSP, 12, 1805–1808, 1987.

Fanucci, L. and Saponara, S.: Data Driven VLSI Computation For Low-Power DCT-Based Video Coding, in: International Conf. on Electronics, Circuits and Systems, 541–544, 2002.

Goetze, J. and Hekstra, G.: An Algorithm and Architecture Based on Orthonormal Micro-Rotations for Computing the Symmetric EVD, in: Integration – The VLSI Journal, 20, 21–39, 1995.

Heyne, B. and Goetze, J.: A Pure Cordic Based FFT for Reconfigurable Digital Signal Processing, in: 12th European Signal Processing Conference, 2004.

Heyne, B., Buecker, M., and Goetze, J.: Implementation of a Cordic Based FFT on a Reconfigurable Hardware Accelerator, in: 3rd Karlsruhe Workshop on Software Radios, 2004.

Hsiao, S., Hu, Y., Juang, T., and Lee, C.: Efficient VLSI Implementations of Fast Multiplierless Approximated DCT Using Parameterized Hardware Modules for Silicon Intellectual Property Design, 52, 1568–1579, 2005.

Jeong, H., Kim, J., and Cho, W. K.: Low-Power Multiplierless DCT Architecture Using Image Correlation, 50, 262–267, 2004.

Li, J. and Lu, S. L.: Low Power Design of Two-Dimensional DCT, in: IEEE Conf. on ASIC and Exhibit, 309–312, 1996.

Loeffler, C., Lightenberg, A., and Moschytz, G. S.: Practical Fast 1-D DCT Algorithms With 11-Multiplications, in: Proc. ICASSP, 2, 988–991, Glasgow,UK, 1989.

Mariatos, E., Metafas, D., Hallas, J., and Goutis, C.: A Fast DCT Processor, Based on Special Purpose CORDIC Rotators, in: IEEE International Symposium on Circuits and Systems, 4, 271–274, 1994.

Richardson, I. E. G.: Video Codec Design, John Wiley & Sons Ltd, Atrium, England, 2002.

Shams, A., Pan, W., Chidanandan, A., and Bayoumi, M. A.: A Low-Power High Performance Distributed DCT Architecture, in: IEEE Computer Society Annual Symposium on VLSI, 21–27, 2002.

Sung, C. C., Ruan, S. J., Lin, B. Y., and Shie, M. C.: Quality and Power Effcient Architecture for the Discrete Cosine Transform, IEICE Transactions on Fundamentals Special Section on VLSI Design and CAD Algorithms, 2005.

Tran, T. D.: A Fast Multiplierless Block Transform for Image and Video Compression, in: International Conf. on Image Processing, 822–826, 1999.

Tran, T. D.: The BinDCT: Fast Multiplierless Approximation of the DCT, 7, 141–144, 2000.

Volder, J.: The CORDIC Trigonometric Computing Technique, IRE Trans. Electron. Comput., EC-8, 330–334, 1959.

Walther, J.: A Unified Algorithm for Elementary Functions, in: Proc. Spring Joint Comput. Conf., 38, 379–385, 1971.

Wang, Z.: Fast Algorithms for the Discrete W Transform and for the Discrete Fourier Transform, 32, 803–816, 1984.