EFFICIENT IMPLEMENTATION OF A CLASS OF MDCT/IMDCT FILTERBANKS FOR SPEECH AND AUDIO CODING APPLICATIONS

Ravi K. Chivukula[*] and *Yuriy A. Reznik* [§,1], *Sr. Member, IEEE*

[*] The University of Texas at Arlington, Electrical Engineering Dept., Arlington, TX 76019
[§]QUALCOMM Incorporated, San Diego, CA 92121

ABSTRACT

In this paper, an efficient algorithm for implementing MDCT/IMDCT of lengths $N = 5 \cdot 2^m$ $(m \geq 2)$ is presented. Transforms for such lengths are of interest for speech and audio coding applications, such as recently issued and/or emerging standards G.729.1, G.EV-VBR, and EVRC-WB. In our design we utilize a mapping of MDCT of size *N* into *N/2*-point DCT-IV and DCT-II with isolated pre-multiplications, which are subsequently moved in the windowing stage. We show that such a modified window is piece-wise symmetric, and can be stored using N/2 words. In our algorithm we also use an efficient factorization of 5-point DCT-II which requires only 4 multiplications by irrational factors. We compare our proposed algorithm with several alternative implementations and show that our design offers practically appreciable reduction in complexity and memory usage.

Index Terms— Speech coding, Audio coding, FFT, DCT, MDCT, Factorization, Fast algorithms.

1. INTRODUCTION

The Modified Discrete Cosine Transform (MDCT) is widely used in speech and audio coding as analysis/synthesis filter bank with time domain alias cancellation property [1]. It usually represents one of the most computationally intensive parts of the codec and hence there exists a need for fast algorithms for implementing it.

This problem is well known, well studied, and numerous efficient algorithms have been proposed for solving it [2-8]. Many of these proposed algorithms are derived for transforms of lengths *N* = *2*^{m}. Among other transform sizes, ones including a factor of 3 (as prompted by the design of MP3 audio codec [9]) have also been thoroughly studied [10-11]. Treatment of transforms of other sizes, however, appears to be somewhat sketchier, and engineers are often left with the necessity to modify or combine some of these techniques in their designs.

The need for transforms of sizes $N = 5 \cdot 2^m$ $(m \geq 2)$, which we will consider in this paper, arises in the design of speech and audio codecs, which typically operate with sampling rates of 8kHz or 16kHz and have to use frames with only 10ms or 20ms delay. Examples of such algorithms include recently standardized ITU-T G.729.1, and 3GPP2 EVRC-WB vocoders [12,13], and an emerging ITU-T G.EV-VBR standard [14].

The algorithm described in this paper offers a simple and practical implementation of transforms, suitable for use in speech and audio coding applications.

The rest of this paper is organized as follows. In Section 2, we explain the mapping of the MDCT into a DCT-IV and DCT-II with isolated pre/post- multiplications, allowing their subsequent absorption by the windowing stage. In Section 3, we describe an efficient implementation of 5-point DCT-II algorithm, adopted in our MDCT design. The design of the merged window is discussed in Section 4. Finally, in Section 5 we characterize the overall complexity of our proposed algorithm and compare it with other possible implementations.

2. MAPPING OF MDCT IN DCT-IV AND DCT-II

Let $\{x(n)\}, n = 0, \ldots, N-1$ represent an input sequence of samples, and *N* denote the frame length. In this paper we consider MDCT and inverse MDCT (IMDCT), defined respectively as follows:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cos\left(\frac{\pi}{2N}\left(2n+1+\frac{N}{2}\right)(2k+1)\right), \quad k = 0, \ldots., \tfrac{N}{2}-1$$

$$\tilde{x}(n) = \sum_{k=0}^{\frac{N}{2}-1} X(k) \cos\left(\frac{\pi}{2N}\left(2n+1+\frac{N}{2}\right)(2k+1)\right), \quad n = 0, \ldots., N-1$$

where *X(k)* are the resulting MDCT coefficients and $\tilde{x}(n)$ are the reconstructed samples. For convenience, we ignore the normalization factors in all definitions.

By adopting the matrix notation proposed in [15], we define the *(N/2*× *N)* MDCT matrix *M*:

$$M(i,j) = \cos\left(\frac{\pi}{2N}\left(2j+1+\frac{N}{2}\right)(2i+1)\right), \qquad \begin{array}{l} i = 0, \ldots, \frac{N}{2}-1, \\ j = 0, \ldots, N-1. \end{array}$$

and write:

[1] Corresponding author. Email: yreznik@ieee.org.

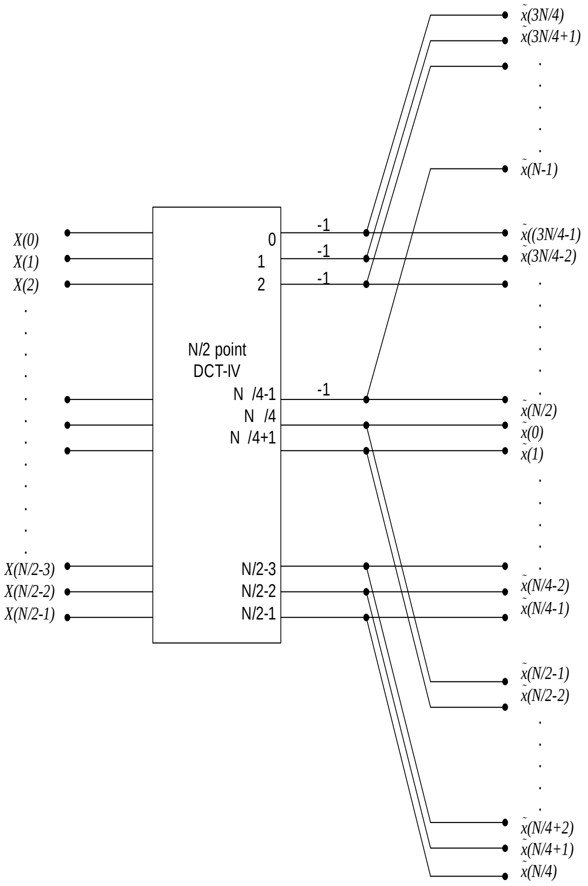

Fig. 1. Implementation of IMDCT using DCT-IV

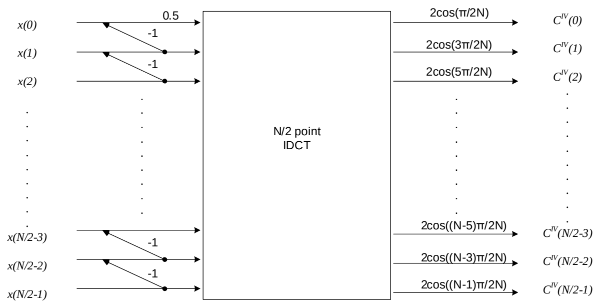

Fig. 2. Implementation of DCT-IV using IDCT-II

$$\boldsymbol{X} = M\boldsymbol{x}$$
$$\tilde{\boldsymbol{x}} = M^T\boldsymbol{X}$$

where:

$$\boldsymbol{X} = \left[X(0),...,X\left(\tfrac{N}{2}-1\right)\right]^T$$
$$\boldsymbol{x} = \left[x(0),...,x(N-1)\right]^T$$
$$\tilde{\boldsymbol{x}} = \left[\tilde{x}(0),...,\tilde{x}(N-1)\right]^T$$

We next map MDCT and IMDCT into an *N/2*-point DCT-IV as follows [15]:

$$M^T = PSC_{N/2}^{IV}$$
$$M = C_{N/2}^{IV}SP^T$$

where:

$$P = \begin{bmatrix} 0 & I_{N/4} \\ 0 & -J_{N/4} \\ J_{N/4} & 0 \\ I_{N/4} & 0 \end{bmatrix},$$

$I_{N/4}$ and $J_{N/4}$ are $N/4\times N/4$ identity and order reversal matrices correspondingly,

$$S = \begin{bmatrix} -I_{N/4} & 0 \\ 0 & I_{N/4} \end{bmatrix},$$

and $C_{N/2}^{IV}$ is an $N/2\times N/2$ DCT-IV matrix, defined as

$$C_{N/2}^{IV}(i,j) = \cos\left(\frac{\pi}{2N}(2i+1)(2j+1)\right),\ \ i,j = 0,...,\tfrac{N}{2}-1.$$

By using the symmetry and involutory properties of the DCT-IV matrix, it can be mapped into DCT-II as follows [16]:

$$C_{N/2}^{IV} = D\left(C_{N/2}^{II}\right)^T L^T$$

where *D* is a diagonal matrix with elements,

$$D(i,i) = 2\cos\left(\frac{\pi}{2N}(2i+1)\right),\ \ i = 0,...,\tfrac{N}{2}-1,$$

$$L = \begin{bmatrix} 0.5 & 0 & 0 & 0 & \cdots & 0 \\ -0.5 & 1 & 0 & 0 & \cdots & 0 \\ 0.5 & -1 & 1 & 0 & \cdots & 0 \\ -0.5 & 1 & -1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -0.5 & 1 & -1 & 1 & \cdots & 1 \end{bmatrix},$$

and $C_{N/2}^{II}$ is an $N/2\times N/2$ DCT-II matrix defined as

$$C_{N/2}^{II}(i,j) = \cos\left(\frac{\pi}{N}(2i+1)j\right),\ \ \ i,j = 0,...,\tfrac{N}{2}-1.$$

We illustrate these transformations in Fig.1 and Fig. 2. It can be seen that the post multiplications contributed by the matrix *D* can be merged with the windowing operation that follows IMDCT. This saves *N/2* multiplications and *N/2* words of memory otherwise used for storing these factors.

The flow graphs for forward transform (MDCT) can be obtained by transposing the flow graphs for IMDCT.

We next are tasked with an efficient implementation of forward and inverse DCT-II. Since we have a factor of 5 in *N*, we require an algorithm that can handle any even length sequence. For this purpose we adopt the algorithm of C.W. Kok [16], which essentially is decimation-in-frequency

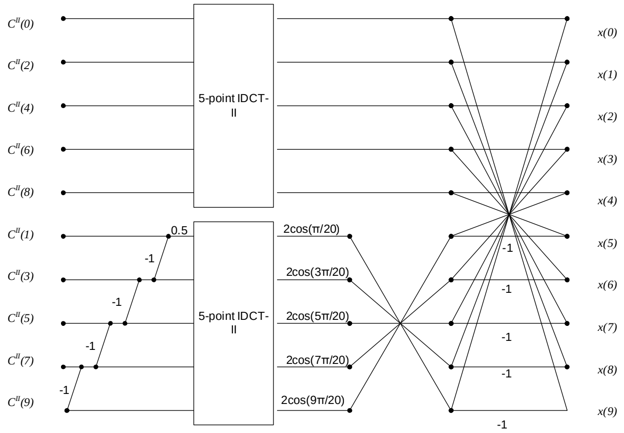

Fig. 3. Implementation of a 10-point inverse DCT-II

strategy and which can be executed recursively until we are left with a DCT of size 5. We present the flow-graph of this process for a transform of size 10 in Fig. 3.

3. FAST 5-POINT DCT-II

As a basis for our 5-point DCT-II we have used an algorithm proposed by M.T. Heideman [17]. We have modified this algorithm by replacing a section with 3 multiplications and 3 additions between coefficients *x(3)* and *x(4)* with the standard planar rotation, and did a similar simplification in the processing path of coefficients *x(0)* and *x(2)*. These changes shorten critical path length, improve structural regularity, and reduce dynamic range of values within the transform.

The flow graph of our forward DCT-II is shown in Fig. 4. The algorithm can be implemented with just 4 non-trivial multiplications and 14 additions. The equations for inverse DCT implementation are given below:

$$\alpha = \frac{1}{4}; \quad \beta = \cos\left(\frac{3\pi}{10}\right); \quad \gamma = -\cos\left(\frac{\pi}{10}\right); \quad \delta = -\frac{\sqrt{5}}{4}$$

$$a_1 = C^{II}(2) + C^{II}(4); \quad a_2 = C^{II}(4) - C^{II}(2)$$

$$b_1 = C^{II}(0) + a_2; \quad b_2 = C^{II}(0) - \alpha a_2$$

$$b_3 = \beta C^{II}(1) + \gamma C^{II}(3); \quad b_4 = -\beta C^{II}(3) + \gamma C^{II}(1)$$

$$b_5 = b_2 + \delta a_1; \quad b_6 = b_2 - \delta a_1$$

$$x(0) = b_6 - b_4; \quad x(1) = b_3 + b_5; \quad x(2) = b_1;$$

$$x(3) = b_5 - b_3; \quad x(4) = b_4 + b_6$$

4. MODIFIED WINDOW FUNCTION

As discussed in Section 2, the pre and post multiplications involved with DCT-IV in MDCT and IMDCT respectively can be merged with the window function. In most practical cases MDCT is used with the following window function (which is commonly referred to as “sine window”):

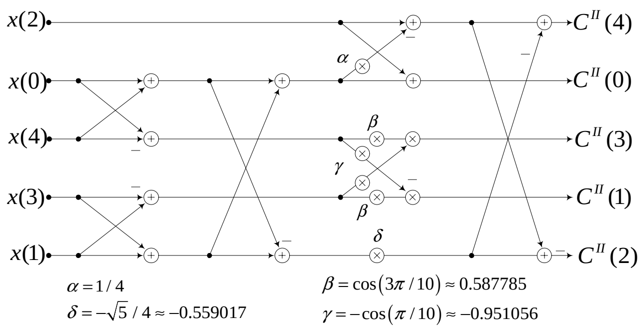

Fig. 4. Five-point forward DCT-II

$$h(n) = \sin\left(\frac{(2n+1)\pi}{2N}\right), \; n = 0, ..., N-1$$

In our case, the combination of transform factors with this window produces the following modified window function:

$$w(n) = \begin{cases} 2\cos\left(\frac{\pi}{2N}\left(2n + \frac{N}{2} + 1\right)\right)\sin\left(\frac{\pi(2n+1)}{2N}\right), & n = 0, ..., \frac{N}{4} - 1, \\ -2\cos\left(\frac{\pi}{2N}\left(2n + \frac{N}{2} + 1\right)\right)\sin\left(\frac{\pi(2n+1)}{2N}\right), & n = \frac{N}{4}, ..., N-1. \end{cases}$$

We note that for $0 \le n < N/4$:

$$w\left(\frac{N}{4} - 1 - n\right) = 2\sin\left(\frac{\pi(2n+1)}{2N}\right)\cos\left(\frac{\pi}{2N}\left(2n + \frac{N}{2} + 1\right)\right)$$
$$= w(n)$$

Similarly, for $0 \le n < 3N/4$:

$$w\left(n + \frac{N}{4}\right) = 2\sin\left(\frac{\pi(2n+1)}{2N}\right)\sin\left(\frac{\pi}{2N}\left(2n + \frac{N}{2} + 1\right)\right)$$
$$= w(N-1-n)$$

Hence the window is piece-wise symmetric. This is illustrated in Fig. 5 (*N* = 640). This means that, compared to the symmetrical sine window, this window can be stored using the same amount of memory.

5. COMPLEXITY ANALYSIS AND COMPARISON

Assume $N = 5.2^m$ ($m \ge 2$) and let $N_1 = 2^m$. Let $RM_F(N)$ and $RA_F(N)$ be, respectively, the multiplicative and additive complexities of an *N*-point MDCT. Let $RM_I(N)$ and $RA_I(N)$ be, respectively, the multiplicative and additive complexities of an *N*-point IMDCT. Let $RM_D(5)$ and $RA_D(5)$ be, respectively, the multiplicative and additive complexities of the fast 5-point DCT-II. Then the computational complexity of the proposed MDCT/IMDCT plus windowing operation is given by

$$RM_F(N) = RM_I(N) = N + \frac{N_1}{2}RM_D(5) + \frac{N(m-1)}{4}$$

$$RA_F(N) = N + \frac{N_1}{2}RA_D(5) + \frac{3N(m-1)}{4} - \frac{N_1}{2}$$

$$RA_I(N) = \frac{N}{2} + \frac{N_1}{2}RA_D(5) + \frac{3N(m-1)}{4} - \frac{N_1}{2}$$

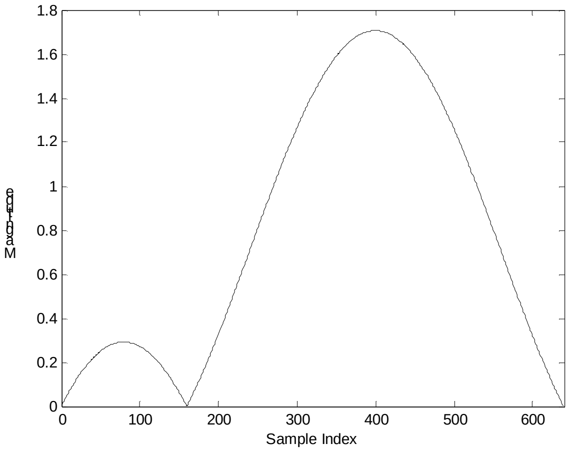

Fig. 5. Modified window function

In Table 1, we present the results of complexity comparison of our algorithm with two possible implementtations of MDCT of size N=640 using the well-known V. Britanak and K.R Rao's algorithm [10]. In first case, we assume that DCT-II of size 5 is implemented using straightforward matrix multiplication, and in the second case, we use same 5-point DCT-II as in our proposed algorithm. The complexity numbers presented in Table 1 include the complexity of windowing operation. This table also presents estimated ROM memory requirements for these algorithms.

6. CONCLUSIONS

We presented an efficient algorithm for MDCT and IMDCT for lengths that are of the type $N = 5.2^m$. This algorithm utilizes several advanced techniques for reduction of complexity and can be of immediate interest for speech and audio coding applications.

7. REFERENCES

[1] Princen, J.; Johnson, A.; Bradley, A., "Subband / Transform coding using filter bank designs based on time domain aliasing cancellation," *in Proc. ICASSP'87.*, vol.12, pp. 2161-2164, Apr 1987.

[2] Malvar, H., "Signal Processing with Lapped Transforms", Artech House, Boston, 1992.

[3] Britanak V., Yip P., Rao K.R., "Discrete Cosine and Sine Transforms: General Properties, Fast Algorithms and Integer Approximations", Academic Press, 2006.

[4] Duhamel, P.; Mahieux, Y.; Petit, J.P., "A fast algorithm for the implementation of filter banks based on `time domain aliasing cancellation'," *in Proc. ICASSP-91,* pp.2209-2212 vol.3, 14-17 Apr 1991.

[5] Digital Audio Compression Standard (AC-3), Rev. B, Doc. A/52B, June 2005.

[6] Fan, Y-H; Madisetti, V.K.; Mersereau, R.M., "On fast algorithms for computing the inverse modified discrete cosine transform," *Signal Processing Letters, IEEE* , vol.6, no.3, pp.61-64, Mar 1999.

Algorithm		Multiplica-tions	Addi-tions	Transform Memory # words	Window Memory # words
MDCT (N=640) with windowing	[10]*	3264	5376	640	320
	[10]**	2176	4992	640	320
	Proposed Algorithm	1856	4352	320	320
IMDCT (N=640) with windowing	[10]*	3264	5056	640	320
	[10]**	2176	4672	640	320
	Proposed Algorithm	1856	4032	320	320

*: without fast 5-point DCT-II, **: with fast 5-point DCT-II

Table 1. Complexity comparison of the proposed algorithm

[7] Chan, D.-Y.; Yang, J.-F.; Chen, S.-Y., "Regular implementation algorithms of time domain aliasing cancellation," *Vision, Image and Signal Processing, IEE Proceedings* - , vol.143, no.6, pp.387-392, Dec 1996.

[8] Chen, C. H.; Wu, C. B.; Liu, B. D.; Yang, J. F.; "Recursive architectures for the forward and inverse modified discrete cosine transforms," *Signal Processing Sytems. IEEE Workshop on*, vol., no., pp.50-59, 2000.

[9] ISO/IEC 13818-3: Information technology — Generic coding of moving pictures and associated audio information — Part 3: Audio, 1998.

[10] Britanak, V.; Rao, K.R., "A new fast algorithm for the unified forward and inverse MDCT/MDST computation" *Signal Processing*, Vol.82, Issue 3, pp.433-459, March 2002.

[11] Shu, H.; Bao, X.; Toumoulin, C.; Luo, L., "Radix-3 Algorithm for the Fast Computation of Forward and Inverse MDCT," *Signal Processing Letters, IEEE* , vol.14, no.2, pp.93-96, Feb. 2007.

[12] 3GPP2 C.S0014-C v1.0, "Enhanced Variable Rate Codec, Speech Service Option 3, 68 and 70 for Wideband Spread Spectrum Digital Systems", 2007.

[13] ITU Recommendation G.722.1, "Low-complexity coding at 24 and 32 kbit/s for hands-free operation in systems with low frame loss", 2005.

[14] ITU-T SG16 Q9 – Contribution 199: Extended high-level description of the Q9 EV-VBR baseline codec, June 2007.

[15] Cheng, M-H.; Hsu, Y-H., "Fast IMDCT and MDCT algorithms - a matrix approach," *Signal Processing, IEEE Transactions on*, vol.51, no.1, pp. 221-229, Jan. 2003.

[16] Kok, C. W., "Fast algorithm for computing discrete cosine transform," *Signal Processing, IEEE Transactions on*, vol.45, no.3, pp.757-760, Mar 1997.

[17] Heideman, M.T., "Computation of an odd-length DCT from a real-valued DFT of the same length," *Signal Processing, IEEE Transactions on*, vol.40, no.1, pp.54-61, Jan 1992.