Implementation of linear-phase FIR nearly perfect reconstruction cosine-modulated filterbanks utilizing the coefficient symmetry

Bregovic, Robert; Yu, Ya Jun; Viholainen, Ari; Lim, Yong Ching

2010

Bregovic, R., Yu, Y. J., Viholainen, A., & Lim, Y. C. (2009). Implementation of linear-phase fir nearly perfect reconstruction cosine-modulated filterbanks utilizing the coefficient symmetry. *IEEE Transactions on Circuits and Systems—I*, 57(1), 139-151.

<https://hdl.handle.net/10356/91329>

<https://doi.org/10.1109/TCSI.2009.2015599>

© 2010 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE. This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder.

Implementation of Linear-Phase FIR Nearly Perfect Reconstruction Cosine-Modulated Filterbanks Utilizing the Coefficient Symmetry

Robert Bregović, *Member, IEEE*, Ya Jun Yu, *Senior Member, IEEE*, Ari Viholainen, *Member, IEEE*, and Yong Ching Lim, *Fellow, IEEE*

Abstract—The analysis and synthesis parts of a cosine-modulated M -channel filterbank (FB) contain two sections, a modulation block and a prototype filter implemented in a polyphase structure. Although, in many cases, a linear-phase prototype filter is used, the coefficient symmetry of this filter is not utilized when using the existing polyphase structure. In this paper, a method is proposed for implementing a linear-phase prototype filter building a nearly perfect reconstruction cosine-modulated FB in such a way that it enables one to partially utilize the coefficient symmetry, thereby reducing the number of required multiplications in the implementation. The proposed method can be applied for implementing FBs with an arbitrary filter order and number of channels. Moreover, it is shown that, in all cases under consideration, the cosine-modulation part of the FB can be implemented by using a fast discrete cosine transform. The efficiency of the proposed implementation is evaluated by means of examples.

Index Terms—Cosine-modulated filterbank (FB), fast discrete cosine transform (DCT), FIR filter, linear-phase, multirate system, nearly perfect reconstruction (NPR).

I. INTRODUCTION

DURING the last three decades, multirate systems have been used in various applications, e.g., adaptive signal processing, compression, denoising, and data transmission [1]–[3]. One of the basic building blocks of a multirate system is the uniform M -channel critically sampled filterbank (FB) shown in Fig. 1. This FB consists of an analysis part (analysis FB), containing filters with the transfer functions $H_k(z)$ for $k = 0, 1, \dots, M-1$ followed by downsamplers by M , and a synthesis part (synthesis FB), containing filters with transfer functions $F_k(z)$ for $k = 0, 1, \dots, M-1$ preceded by up-samplers by M . Moreover, in this paper, it is assumed that the processing unit does not modify the subband signals.

Manuscript received August 05, 2008; revised November 12, 2008. First published February 18, 2009; current version published January 27, 2010. This work was supported by Temasek Laboratories, Nanyang Technological University, Singapore, and by the Academy of Finland under Project 201019 (post-doctoral research grant), and 213462 [Finnish Centre of Excellence program (2006–2011)]. This paper was recommended by Associate Editor H. Johansson.

R. Bregović is with the Department of Signal Processing, Tampere University of Technology (TUT), 33101 Tampere, Finland (e-mail: bregovic@cs.tut.fi).

Y. J. Yu and Y. C. Lim are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: eleyuyj@gmail.ntu.edu.sg; elelimyc@gmail.ntu.edu.sg).

A. Viholainen is with the Department of Communications Engineering, Tampere University of Technology (TUT), 33101 Tampere, Finland (e-mail: ari.viholainen@tut.fi).

Digital Object Identifier 10.1109/TCSI.2009.2015599

Fig. 1. M -channel critically sampled FB.

When synthesizing an FB, the goal is to generate a system that has either a perfect reconstruction (PR) or a nearly perfect reconstruction (NPR) property. In the PR case, the output signal $y[n]$ is a delayed version of the input signal $x[n]$, i.e., $y[n] = x[n - D]$ with D being the FB delay. In the NPR case, this relation is only approximately satisfied, i.e., $y[n] \approx x[n - D]$. By giving up on the PR property, FBs with better overall performance can be designed, for example, the same channel selectivity can be achieved by having filters of a lower order [4]. Therefore, for most applications, systems satisfying the NPR property are a better choice as long as the distortions introduced by the FB are smaller than the changes caused by the application.

Among the different types of M -channel FBs, the most commonly used ones are modulated FBs. In a modulated FB, the filter transfer functions $H_k(z)$ and $F_k(z)$ are generated by properly modulating one prototype filter.¹ In this case, only one filter has to be designed, and the implementation consists of a polyphase implementation of the prototype filter and a modulation block, thereby simplifying the design and implementation of M -channel FBs.

This paper considers the implementation of cosine-modulated FBs² with linear-phase FIR prototype filters. Cosine-modulated FBs are modulated FBs with modulation matrices based on cosine functions. In PR cosine-modulated FBs with linear-phase prototype filters of order $N = 2KM - 1$, with K being an integer, the prototype filters can be efficiently implemented by utilizing a lattice structure [1]–[3]. In such FBs, the implementation of the prototype filter requires $(N + 1)/2$ multiplications per M input samples. However, this implementation cannot be used for NPR FBs. For NPR cosine-modulated

¹In some modulated FBs, different prototype filters are used for generating the analysis and synthesis FBs. In this paper, without loss of generality, FBs with only one prototype filter are considered.

²For processing real-valued signals, cosine-modulated FBs are one of the more frequently used types of modulated FBs.

FBs with arbitrary N and M values, the currently most efficient implementation for the prototype filter is achieved by using a polyphase structure. In this case, the implementation of the prototype filter requires $N + 1$ multiplications per M input samples. The drawback of this implementation is that the coefficient symmetry of the linear-phase prototype filter is not utilized, i.e., the implementation complexity of the prototype filter is the same for linear-phase as well as non-linear-phase filters.

It has been shown in [5] that, for NPR FBs with prototype filters of order $N = 2KM - 1$, the coefficient symmetry of the prototype filters can be partially utilized. However, this order selection is very restrictive due to the fact that, in the case of NPR FBs, prototype filters of any order can be used compared to the PR case where only filters of order $N = 2KM - 1$ result in good FBs [4], [6], [7] (this is also illustrated by means of an example in Section VI). Therefore, in this paper, a method is proposed for implementing a linear-phase prototype filter of an arbitrary order in such a way that it enables one to partially utilize the coefficient symmetry, thereby reducing the number of required multiplications in the implementation. Furthermore, it is shown that, in all cases under consideration, the cosine-modulation part of the FB can be implemented by using a fast discrete cosine transform (DCT). In order to simplify the discussion, in this paper, only FBs with even numbers of channels and odd filter orders are considered. Similar principles presented in this paper can be applied to other cases.

The outline of this paper is as follows. Section II reviews the basic relations and properties of cosine-modulated FBs. This section also shows how the cosine-modulation part of the FB can be efficiently implemented by utilizing a fast DCT (the proof is given in Appendix A). The implementation of the polyphase part of the FB is discussed in Section III. The proposed implementation method is given in Section IV with the expressions for evaluating the implementation complexity given in Section V. In Section VI, a comparison is performed between the proposed implementation method and the polyphase one. Finally, some concluding remarks are given in Section VII.

II. COSINE-MODULATED FBs

This section reviews the basic properties and implementation structures for cosine-modulated FBs with synthesis and analysis filters derived by modulating one linear-phase FIR prototype filter. Moreover, the emphasis is put on FBs with even numbers of channels and odd filter orders. It is also shown that, for all cases under consideration, the cosine-modulation part can be implemented by using a fast DCT.

For a linear-phase prototype filter of order N with the transfer function

$$H(z) = \sum_{n=0}^N h[n]z^{-n} \quad (1)$$

where $h[N - n] = h[n]$ for $n = 0, 1, \dots, N$, the impulse-response coefficients of filters with the transfer functions $H_k(z)$ and $F_k(z)$ building an M -channel cosine-modulated FB are generated by using the following modulation functions [8]–[10]:

$$h_k[n] = 2h[n] \cos \left[\left(k + \frac{1}{2} \right) \frac{\pi}{M} \left(n - \frac{D}{2} \right) + (-1)^k \frac{\pi}{4} \right] \quad (2a)$$

$$f_k[n] = 2h[n] \cos \left[\left(k + \frac{1}{2} \right) \frac{\pi}{M} \left(n - \frac{D}{2} \right) - (-1)^k \frac{\pi}{4} \right] \quad (2b)$$

for $k = 0, 1, \dots, M - 1$ and $n = 0, 1, \dots, N$. In the aforementioned equation, D denotes the FB delay. For a cosine-modulated FB with a linear-phase prototype filter, the FB delay is equal to the filter order, i.e., $D = N$. Moreover, as the emphasis in this paper is put on FBs with even numbers of channels and odd filter orders, the filter order will be represented as

$$N = 2K_E M + 2\Delta - 1 \quad (3a)$$

with K_E being an even integer, Δ being an integer, and $0 \leq \Delta \leq 2M - 1$. Consequently

$$K_E = 2 \left\lfloor \frac{N+1}{4M} \right\rfloor \quad (3b)$$

$$\Delta = \frac{N+1}{2} - K_E M. \quad (3c)$$

Parameters K_E and Δ will be used later on when deriving the implementation structures.

As briefly mentioned in Section I, cosine-modulated FBs have the following two main properties: First, instead of designing M analysis and M synthesis filters, only one prototype filter has to be designed, thereby significantly simplifying the FB design. Second, the FB can be implemented as shown in Fig. 2. This implementation consists of the prototype filter implemented in its polyphase form and a cosine-modulation matrix. The polyphase terms are generated by decomposing the prototype filter given by (1) into $2M$ polyphase components as

$$H(z) = \sum_{\xi=0}^{2M-1} z^{-\xi} G_{\xi}(z^{2M}) \quad (4a)$$

where $G_{\xi}(z)$ for $\xi = 0, 1, \dots, 2M - 1$ is the ξ th polyphase component defined as

$$G_{\xi}(z) = \sum_{n=0}^{N_{\xi}} h[\xi + 2Mn]z^{-n} \quad (4b)$$

with N_{ξ} being the order of the ξ th polyphase component defined as

$$N_{\xi} = \left\lfloor \frac{N+1-\xi}{2M} \right\rfloor - 1. \quad (4c)$$

The elements of the cosine-modulation matrices $\mathbf{C}_1$ and $\mathbf{C}_2$ are given by the following [8]–[10]:

$$[\mathbf{C}_1]_{kl} = 2 \cos \left(\frac{\pi}{M} \left(k + \frac{1}{2} \right) \left(l - \frac{N}{2} \right) + (-1)^k \frac{\pi}{4} \right) \quad (5a)$$

$$[\mathbf{C}_2]_{kl} = 2 \cos \left(\frac{\pi}{M} \left(k + \frac{1}{2} \right) \left(2M - 1 - l - \frac{N}{2} \right) - (-1)^k \frac{\pi}{4} \right) \quad (5b)$$

for $k = 0, 1, \dots, M - 1$ and $l = 0, 1, \dots, 2M - 1$. Due to the fact that the synthesis FB is implemented in a similar way as the analysis one, in the rest of this paper, only the analysis FB is considered.

Fig. 2. Polyphase structure for implementing a cosine-modulated FB.

For FBs with prototype filters of order $N = 2K_E M + 2\Delta - 1$, with K_E and M being even integers and $0 \leq \Delta \leq 2M - 1$, the implementation structure shown in Fig. 2 can further be simplified to the one shown in Fig. 3. In this figure, $G(z)$ represents the polyphase implementation of the prototype filter, as shown in Fig. 4, and the cosine-modulation matrix $\mathbf{C}_1$ from Fig. 2 has been decomposed into four parts as

$$\mathbf{C}_1 = \lambda \mathbf{C}_{\text{DCT}}^{(\text{IV})} \mathbf{T} \mathbf{S}_{\Delta}^{(2M)}. \quad (6a)$$

Here, λ is the scaling factor defined as

$$\lambda = (-1)^{K_E/2} \sqrt{M}. \quad (6b)$$

$\mathbf{C}_{\text{DCT}}^{(\text{IV})}$ is the DCT-IV transform defined as [1], [11]

$$[\mathbf{C}_{\text{DCT}}^{(\text{IV})}]_{k,l} = \sqrt{\frac{2}{M}} \cos \left[\frac{\pi}{M} \left(k + \frac{1}{2} \right) \left(l + \frac{1}{2} \right) \right] \quad (6c)$$

for $k, l = 0, 1, \dots, M-1$. Matrix $\mathbf{T}$ of size M by $2M$ is defined as

$$\mathbf{T} = [\mathbf{I}_M - \mathbf{J}_M \quad -(\mathbf{I}_M + \mathbf{J}_M)] \quad (6d)$$

with $\mathbf{I}_M$ and $\mathbf{J}_M$ being the identity and counter-identity matrices of size M by M . Moreover, matrix $\mathbf{T}$ is further decomposed as

$$\mathbf{T} = \mathbf{T}_B \mathbf{T}_A \quad (6e)$$

with

$$\mathbf{T}_A = \begin{bmatrix} \mathbf{I}_{M/2} & -\mathbf{J}_{M/2} & \mathbf{0}_{M/2} & \mathbf{0}_{M/2} \\ \mathbf{0}_{M/2} & \mathbf{0}_{M/2} & \mathbf{J}_{M/2} & \mathbf{I}_{M/2} \end{bmatrix} \quad (6f)$$

$$\mathbf{T}_B = \begin{bmatrix} \mathbf{I}_{M/2} & -\mathbf{J}_{M/2} \\ -\mathbf{J}_{M/2} & -\mathbf{I}_{M/2} \end{bmatrix} \quad (6g)$$

and $\mathbf{0}_{M/2}$ being a zero matrix of size $M/2$ by $M/2$. Finally, matrix $\mathbf{S}_{\Delta}^{(2M)}$ of size $2M$ by $2M$ has the following form:

$$\mathbf{S}_{\Delta}^{(2M)} = \begin{bmatrix} \mathbf{0}_{2M-\Delta, \Delta} & \mathbf{I}_{2M-\Delta} \\ -\mathbf{I}_{\Delta} & \mathbf{0}_{\Delta, 2M-\Delta} \end{bmatrix} \quad (6h)$$

with $\mathbf{0}_{2M-\Delta, \Delta}$ being a zero matrix of size $2M - \Delta$ by Δ and Δ being defined by (3c). The proof for (6a)–(6h) can be found in Appendix A.

Based on (6h), it can be seen that matrix $\mathbf{S}_{\Delta}^{(2M)}$ is only a cross-connection network that connects outputs of the polyphase fil-

Fig. 3. Efficient polyphase structure for implementing an analysis FB with $N = 2K_E M + 2\Delta - 1$.

Fig. 4. Polyphase implementation of the prototype filter.

Fig. 5. Implementation of the cross-connection matrix $\mathbf{S}_{\Delta}^{(2M)}$.

ters with the inputs of matrix $\mathbf{T}$ in a one-to-one basis. Therefore, it is straightforward to implement matrix $\mathbf{S}_{\Delta}^{(2M)}$, as shown in Fig. 5. Consequently, the relation between $\tilde{x}_k[m]$ and $v_k[m]$ is given by

$$v_k[m] = \begin{cases} \tilde{x}_{k+\Delta}[m], & \text{for } k = 0, \dots, 2M - \Delta - 1 \\ -\tilde{x}_{k+\Delta-2M}[m], & \text{for } k = 2M - \Delta, \dots, 2M - 1. \end{cases} \quad (7)$$

TABLE I
EQUATIONS DEFINING QUADRUPLTS USED FOR IMPLEMENTING THE FB

Δ	Quadruplets
$0 \leq \Delta \leq \frac{M}{2} - 1$	$\begin{bmatrix} U_\mu(z) \\ U_{M-\mu-1}(z) \end{bmatrix} = \begin{bmatrix} G_{\mu+\Delta}(-z^2) & -G_{M-\mu+\Delta-1}(-z^2) \\ z^{-1}G_{\mu+\Delta+M}(-z^2) & z^{-1}G_{2M-\mu+\Delta-1}(-z^2) \end{bmatrix} \begin{bmatrix} X_{\mu+\Delta}(z) \\ X_{M-\mu+\Delta-1}(z) \end{bmatrix} \text{ for } \mu = \Delta+1, \dots, \frac{M}{2}-1 \quad (9a)$
$1 \leq \Delta \leq M-1$	$\begin{bmatrix} U_\mu(z) \\ U_{M-\mu-1}(z) \end{bmatrix} = \begin{bmatrix} G_{\mu+\Delta}(-z^2) & -z^{-1}G_{M-\mu+\Delta-1}(-z^2) \\ z^{-1}G_{\mu+\Delta+M}(-z^2) & -G_{\Delta-\mu-1}(-z^2) \end{bmatrix} \begin{bmatrix} X_{\mu+\Delta}(z) \\ X_{\Delta-\mu-1}(z) \end{bmatrix} \text{ for } \mu = 0, 1, \dots, \frac{M}{2}-1 - \left\lfloor \Delta - \frac{M}{2} \right\rfloor \quad (9b)$
$\frac{M}{2}+1 \leq \Delta \leq \frac{3M}{2}-1$	$\begin{bmatrix} U_\mu(z) \\ U_{M-\mu-1}(z) \end{bmatrix} = \begin{bmatrix} z^{-1}G_{\mu+\Delta}(-z^2) & -z^{-1}G_{M-\mu+\Delta-1}(-z^2) \\ -G_{\mu+\Delta-M}(-z^2) & -G_{\Delta-\mu-1}(-z^2) \end{bmatrix} \begin{bmatrix} X_{\mu+\Delta-M}(z) \\ X_{\Delta-\mu-1}(z) \end{bmatrix} \text{ for } \mu = \Delta-M , \Delta-M +1, \dots, \frac{M}{2}-1 \quad (9c)$
$M+1 \leq \Delta \leq 2M-1$	$\begin{bmatrix} U_\mu(z) \\ U_{M-\mu-1}(z) \end{bmatrix} = \begin{bmatrix} z^{-1}G_{\mu+\Delta}(-z^2) & G_{\Delta-\mu-M-1}(-z^2) \\ -G_{\mu+\Delta-M}(-z^2) & -z^{-1}G_{\Delta-\mu-1}(-z^2) \end{bmatrix} \begin{bmatrix} X_{\mu+\Delta-M}(z) \\ X_{\Delta-\mu-1}(z) \end{bmatrix} \text{ for } \mu = 0, 1, \dots, \frac{M}{2}-1 - \left\lfloor \Delta - \frac{3M}{2} \right\rfloor \quad (9d)$
$\frac{3M}{2}+1 \leq \Delta \leq 2M-1$	$\begin{bmatrix} U_\mu(z) \\ U_{M-\mu-1}(z) \end{bmatrix} = \begin{bmatrix} -G_{\mu+\Delta-2M}(-z^2) & G_{\Delta-\mu-M-1}(-z^2) \\ -z^{-1}G_{\mu+\Delta-M}(-z^2) & -z^{-1}G_{\Delta-\mu-1}(-z^2) \end{bmatrix} \begin{bmatrix} X_{\mu+\Delta-2M}(z) \\ X_{\Delta-\mu-M-1}(z) \end{bmatrix} \text{ for } \mu = 2M-\Delta, 2M-\Delta+1, \dots, \frac{M}{2}-1 \quad (9e)$

Fig. 6. Implementation for one set of matrix T .

For FBs with $N = 2K_E M - 1$, with K_E and M being even integers, $v_k[m] = \tilde{x}_k[m]$ for $k = 0, 1, \dots, 2M - 1$, i.e., $S_\Delta^{(2M)} = I_{2M}$.

The matrix T given by (6d) combines the outputs of four polyphase filters by adding or subtracting them in order to build two inputs to the DCT-IV. As there are $2M$ polyphase filters, there are altogether $M/2$ such sets. Each of these sets is defined by the following:

$$\begin{bmatrix} w_\mu[m] \\ w_{M-\mu-1}[m] \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} u_\mu[m] \\ u_{M-\mu-1}[m] \end{bmatrix} \\ = \begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} v_\mu[m] \\ v_{M-\mu-1}[m] \\ v_{\mu+M}[m] \\ v_{2M-\mu-1}[m] \end{bmatrix} \quad (8)$$

for $\mu = 0, 1, \dots, M/2 - 1$. The implementation of (8) is shown in Fig. 6.

There are four important observations related to the modulation part that can be seen from (6a)– to (6h), as well as Fig. 6. First, matrix T (as well as T_A and T_B) contains only additions (subtractions) and, as such, can be implemented straightforwardly. Second, the DCT-IV transform given by (6c) is of size M by M . This is half the original modulation matrix of size M by $2M$. Third, matrices T and $C_{\text{DCT}}^{(\text{IV})}$ depend only on the number of channels and do not depend on the filter order and, consequently, FB delay. Therefore, when deriving an efficient implementation for the FB, the part related to the DCT-IV does not have to be considered as it is identical for all FBs having the

TABLE II
SELECTION OF (9a)–(9e) DEPENDING ON THE VALUE OF Δ

Δ	(9a)	(9b)	(9c)	(9d)	(9e)
$\Delta=0$	+				
$1 \leq \Delta \leq M/2-1$	+	+			
$\Delta=M/2$		+			
$M/2+1 \leq \Delta \leq M-1$		+	+		
$\Delta=M$			+		
$M+1 \leq \Delta \leq 3M/2-1$			+	+	
$\Delta=3M/2$				+	
$3M/2+1 \leq \Delta \leq 2M-1$				+	+

same number of channels. Fourth, the DCT-IV is a well-known transform that can be efficiently implemented by using a fast DCT [11], [12].

III. IMPLEMENTATION OF THE POLYPHASE SECTION OF THE FB

By combining the equations given in the previous section, for FBs with even numbers of channels M and odd filter orders N , the relations between two inputs of the modulation matrix $C_{\text{DCT}}^{(\text{IV})}$ and two inputs of the corresponding four polyphase components can be expressed in the z domain, depending on Δ , by (9a)–(9e) (see Table I),³ with

$$\begin{bmatrix} W_\mu(z) \\ W_{M-\mu-1}(z) \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} U_\mu(z) \\ U_{M-\mu-1}(z) \end{bmatrix} \quad (10)$$

and Δ given by (3c). Here, $X(z)$, $U(z)$, and $W(z)$ are the z -transforms of $x[m]$, $u[m]$, and $w[m]$, respectively. In the rest of this paper, each of these sets of four polyphase components is referred to as a quadruplet. It should be pointed out that, in an FB, quadruplets from one or two of the aforementioned sets have to be used. This is illustrated in Table II.

In order to illustrate the relations (9a)–(9e), as an example, the implementation of an FB with $M = 8$ and $N = 33$ is considered. In this case, according to (3a) and (3c), $\Delta = 1$ and

³It should be pointed out that the relations given by (9a)–(9e) (see Table I) are somehow similar to the ones derived in [9] for describing the PR requirements. However, the ones introduced here are more useful for deriving an efficient implementation because all signals are organized in such a way that they can be directly used as inputs into a DCT-IV. No further rearrangements are needed.

Fig. 7. Implementation structure for quadruplets given by relations (9a)–(9e).

$K_E = 2$. According to (9a)–(9e), there are the following four sets of equations:

$$\Delta = 1, \mu = 0 \rightarrow (9b) \rightarrow \begin{bmatrix} U_0(z) \\ U_7(z) \end{bmatrix} = \begin{bmatrix} G_1(-z^2) & -z^{-1}G_8(-z^2) \\ z^{-1}G_9(-z^2) & -G_0(-z^2) \end{bmatrix} \begin{bmatrix} X_1(z) \\ X_0(z) \end{bmatrix} \quad (11a)$$

$$\Delta = 1, \mu = 1 \rightarrow (9a) \rightarrow \begin{bmatrix} U_1(z) \\ U_6(z) \end{bmatrix} = \begin{bmatrix} G_2(-z^2) & -G_7(-z^2) \\ z^{-1}G_{10}(-z^2) & z^{-1}G_{15}(-z^2) \end{bmatrix} \begin{bmatrix} X_2(z) \\ X_7(z) \end{bmatrix} \quad (11b)$$

$$\Delta = 1, \mu = 2 \rightarrow (9a) \rightarrow \begin{bmatrix} U_2(z) \\ U_5(z) \end{bmatrix} = \begin{bmatrix} G_3(-z^2) & -G_6(-z^2) \\ z^{-1}G_{11}(-z^2) & z^{-1}G_{14}(-z^2) \end{bmatrix} \begin{bmatrix} X_3(z) \\ X_6(z) \end{bmatrix} \quad (11c)$$

$$\Delta = 1, \mu = 3 \rightarrow (9a) \rightarrow \begin{bmatrix} U_3(z) \\ U_4(z) \end{bmatrix} = \begin{bmatrix} G_4(-z^2) & -G_5(-z^2) \\ z^{-1}G_{12}(-z^2) & z^{-1}G_{13}(-z^2) \end{bmatrix} \begin{bmatrix} X_4(z) \\ X_5(z) \end{bmatrix}. \quad (11d)$$

Fig. 8. Implementation structure for quadruplets under consideration. (a) Type 1. (b) Type 2.

The implementations for one quadruplet given by (9a)–(9e) are shown in Fig. 7(a)–(7e), respectively. As every FB under consideration can be decomposed by using these relations, the goal now is to derive efficient implementations for these five types of quadruplets. This is shown in the next section.

IV. PROPOSED IMPLEMENTATION METHOD

In order to derive an efficient implementation for the five quadruplets given in the previous section and consequently derive an efficient implementation for the FBs under consideration, it should be observed from Fig. 7(a)–(7e) that there are, in principle, only two different quadruplets to be implemented. Namely, the quadruplets shown in Fig. 7(a), 7(c), and 7(e) are similar from the implementation point of view.⁴ The same applies for the ones shown in Fig. 7(b) and 7(d). Therefore, in this paper, only the implementation for quadruplets given in Fig. 7(a) and 7(b) is considered.

In order to simplify the notations, in Fig. 8,⁵ the two quadruplets under consideration are shown again with the transfer functions $G_\chi(-z^2)$ from Fig. 7(a)–(7e) replaced by generic transfer functions $A(-z^2)$, $B(-z^2)$, $C(-z^2)$, and $D(-z^2)$ defined as

$$A(z) = \sum_{n=0}^{R_0} a_n z^{-n} \quad (12a)$$

$$B(z) = \sum_{n=0}^{R_1} b_n z^{-n} \quad (12b)$$

$$C(z) = \sum_{n=0}^{R_1} c_n z^{-n} \quad (12c)$$

$$D(z) = \sum_{n=0}^{R_0} d_n z^{-n} \quad (12d)$$

⁴For example, the structure in Fig. 7(e) is actually the structure in Fig. 7(a) multiplied with -1 . Consequently, the structure in Fig. 7(e) can be implemented as the structure in Fig. 7(a) with the input or output signals multiplied by -1 .

⁵The part related to (10) is omitted from these figures as that part consists only from two additions.

TABLE III
FILTER ORDERS FOR GIVEN VALUES OF Δ

Δ	Type 1		Type 2		
	R_0, R_1	R	R_0	R_1	R
$\Delta=0$	K_E-1	K_E-1	$\times$	$\times$	$\times$
$1 \leq \Delta \leq M/2-1$	K_E-1	K_E-1	$\times$	$\times$	$\times$
$\Delta=M/2$	$\times$	$\times$	K_E	K_E-1	K_E
$M/2+1 \leq \Delta \leq M-1$	$\times$	$\times$	$\times$	$\times$	$\times$
$\Delta=M$	K_E	K_E	$\times$	$\times$	$\times$
$M+1 \leq \Delta \leq 3M/2-1$	$\times$	$\times$	$\times$	$\times$	$\times$
$\Delta=3M/2$	$\times$	$\times$	K_E+1	K_E	K_E+1
$3M/2+1 \leq \Delta \leq 2M-1$	K_E+1	K_E+1	$\times$	$\times$	$\times$

with R_0 and R_1 being the filter orders of these generic transfer functions. These orders are calculated according to (4c) and are related to the parameter K_E , based on Δ , as shown in Table III. Parameter R given in the table will be used later on. The relation between orders R_0 and R_1 and parameters K_E and Δ could also be expressed by using equations but the table is more self-explanatory. Moreover, due to the linear-phase property of the prototype filter, it turns out that, for all cases under consideration

$$d_n = a_{R_0-n}, \quad \text{for } n = 0, 1, \dots, R_0 \quad (13a)$$

$$c_n = b_{R_1-n}, \quad \text{for } n = 0, 1, \dots, R_1. \quad (13b)$$

A proof for the aforementioned relations is given in Appendix B. As an example, an FB with $M = 8$ and $N = 33$ is considered. For $\mu = 0$, the quadruplet given by (9b) [see also (11a)] contains polyphase components $\mathbf{g}_0 = [h_0 \ h_{16} \ h_1]$, $\mathbf{g}_9 = [h_9 \ h_8]$, $\mathbf{g}_8 = [h_8 \ h_9]$, and $\mathbf{g}_1 = [h_1 \ h_{16} \ h_0]$ corresponding to filters with transfer functions $A(z)$, $B(z)$, $C(z)$, and $D(z)$, respectively. For this case, the filter orders are $R_0 = 2$ and $R_1 = 1$. It is obvious that the symmetries given by (13a) and (13b) hold. The same can be shown for all other quadruplets.

In the next two sections, it is shown how to efficiently implement each of those quadruplets.

A. Type 1

A Type 1 quadruplet, as shown in Fig. 8(a), is defined by the following input-output relation:

$$\begin{bmatrix} U_0(z) \\ U_1(z) \end{bmatrix} = \begin{bmatrix} A(-z^2) & -C(-z^2) \\ z^{-1}B(-z^2) & z^{-1}D(-z^2) \end{bmatrix} \begin{bmatrix} X_0(z) \\ X_1(z) \end{bmatrix}. \quad (14)$$

In the case under consideration (see Table III)⁶

$$R_0 = R_1 = K_E - 1. \quad (15a)$$

⁶As seen in Table III, for the quadruplets shown in Fig. (9c) and (9e), $R_0 = R_1$ is equal to K_E and $K_E + 1$, respectively.

In order to keep the equations compact, in the rest of this section, R will be used instead of $K_E - 1$, i.e.,

$$R = K_E - 1. \quad (15b)$$

Moreover, as K_E is even, R is an odd integer. By taking (15a) and (15b) into account, (14) can be written in the time domain as shown in (16a) at the bottom of the page. After applying the coefficient symmetries defined by (13a) and (13b), the time domain representation is given by (16b), as shown at the bottom of the page, with

$$\mathbf{x}_{k,\eta,l}^{(t)}[m] = \begin{bmatrix} x_t[m-k] \\ x_t[m-k-\eta] \\ x_t[m-k-2\eta] \\ \vdots \\ x_t[m-k-l\eta] \end{bmatrix} \quad (17)$$

for $m, k, l, \eta \in \mathbb{Z}$ and $t = 0, 1$. In order to derive an efficient implementation, (16b) can be rewritten as

$$\begin{bmatrix} u_0[m] \\ u_1[m+1] \end{bmatrix} = \begin{bmatrix} a_0 & -a_1 & \dots & a_{R-1} & -a_R & -b_R & b_{R-1} & \dots & -b_1 & b_0 \\ b_0 & -b_1 & \dots & b_{R-1} & -b_R & a_R & -a_{R-1} & \dots & a_1 & -a_0 \end{bmatrix} \times \begin{bmatrix} \mathbf{x}_{0,2,R}^{(0)}[m] \\ \mathbf{x}_{0,2,R}^{(1)}[m] \end{bmatrix}. \quad (18)$$

Here, the delays z^{-1} in (14) have been moved to the left side of the equation, based on the following set of identities:

$$U(z) = z^{-1}X(z) \Leftrightarrow u[m] = x[m-1] \Leftrightarrow u[m+1] = x[m]. \quad (19)$$

Therefore, in (18), $u_1[m+1]$ instead of $u_1[m]$ has been evaluated. Moreover, the columns containing only zeros have been removed.

By utilizing the efficient way of doing the complex multiplications (see, for example, [13]), the systems given by (18) can be efficiently implemented as shown in (20a) at the bottom of the next page, with

$$e_0 = a_0 + b_0, e_1 = -a_1 - b_1, \dots, e_R = -a_R - b_R \quad (20b)$$

$$f_0 = a_0 - b_0, f_1 = -a_1 + b_1, \dots, f_R = -a_R + b_R \quad (20c)$$

$$\mathbf{x}_{k_0,\eta_0,l_0}^{k_1,\eta_1,l_1,-}[m] = \begin{bmatrix} x_0[m-k_0] - x_1[m-k_1-l_1\eta_1] \\ x_0[m-k_0-\eta_0] - x_1[m-k_1-(l_1-1)\eta_1] \\ \vdots \\ x_0[m-k_0-l_0\eta_0] - x_1[m-k_1] \end{bmatrix}. \quad (20d)$$

$$\begin{bmatrix} u_0[m] \\ u_1[m] \end{bmatrix} = \begin{bmatrix} a_0 & 0 & -a_1 & 0 & \dots & 0 & -a_R & 0 & -c_0 & 0 & c_1 & 0 & \dots & c_{R-1} & 0 \\ 0 & b_0 & 0 & -b_1 & \dots & b_{R-1} & 0 & -b_R & 0 & d_0 & 0 & -d_1 & \dots & 0 & -d_R \end{bmatrix} \begin{bmatrix} \mathbf{x}_{0,1,2R+1}^{(0)}[m] \\ \mathbf{x}_{0,1,2R+1}^{(1)}[m] \end{bmatrix} \quad (16a)$$

$$\begin{bmatrix} u_0[m] \\ u_1[m] \end{bmatrix} = \begin{bmatrix} a_0 & 0 & -a_1 & 0 & \dots & 0 & -a_R & 0 & -b_R & 0 & b_{R-1} & 0 & \dots & b_0 & 0 \\ 0 & b_0 & 0 & -b_1 & \dots & b_{R-1} & 0 & -b_R & 0 & a_R & 0 & -a_{R-1} & \dots & 0 & -a_0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{0,1,2R+1}^{(0)}[m] \\ \mathbf{x}_{0,1,2R+1}^{(1)}[m] \end{bmatrix} \quad (16b)$$

The coefficients e_k and f_k depend only on the prototype filter coefficients and, as such, can be precalculated. Furthermore, it should be pointed out that, for the case with $N = 2KM - 1$, all quadruplets in the FB are of Type 1. Therefore, in this special case, i.e., for $N = 2KM - 1$, the proposed implementation is similar to the one reported in [5].

As an example, the implementation for a Type 1 quadruplet for $M = 8$ and $N = 33$ is considered. As can be seen from (9a)–(9e), there are three Type 1 quadruplets in this FB generated for $\mu = 1, 2, 3$ (see also (11b)–(11d)). The one for $\mu = 1$ is considered here. After applying the aforementioned derivation on the quadruplet given by (9a), the following system of equations is obtained:

$$\begin{bmatrix} u_1[m] \\ u_6[m+1] \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} e_0 & e_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -h_7 & h_{10} & 0 & 0 \\ 0 & 0 & 0 & 0 & f_1 & f_0 \end{bmatrix} \times \begin{bmatrix} x_2[m] \\ x_2[m-2] \\ x_2[m] - x_7[m-2] \\ x_2[m-2] - x_7[m] \\ x_7[m] \\ x_7[m-2] \end{bmatrix} \quad (21a)$$

with

$$e_0 = h_2 + h_7 \quad e_1 = -h_{15} - h_{10} \quad (21b)$$

$$f_0 = h_2 - h_7 \quad f_1 = -h_{15} + h_{10}. \quad (21c)$$

The implementation structure is shown in Fig. 9. A similar implementation can also be derived for $\mu = 2$ and $\mu = 3$.

B. Type 2

A Type 2 quadruplet, as shown in Fig. 8(b), is defined by the following input–output relation:

$$\begin{bmatrix} U_0(z) \\ U_1(z) \end{bmatrix} = \begin{bmatrix} A(-z^2) & -z^{-1}C(-z^2) \\ z^{-1}B(-z^2) & -D(-z^2) \end{bmatrix} \begin{bmatrix} X_0(z) \\ X_1(z) \end{bmatrix}. \quad (22)$$

In the case under consideration (see Table III),⁷

$$R_0 = K_E \quad (23a)$$

$$R_1 = K_E - 1. \quad (23b)$$

The difference between the orders R_0 and R_1 is due to a non-complete polyphase decomposition ($N + 1 \neq 2KM$ with K

⁷As seen in Table III, for the quadruplet shown in Fig. 9(d), R_0 and R_1 are equal to $K_E + 1$ and K_E , respectively

being an integer), i.e., some polyphase terms have more coefficients than others. Again, in this section, R instead of K_E will be used, i.e.,

$$R = K_E. \quad (23c)$$

By taking into account (23a)–(23c) as well as the coefficient symmetries given by (13a) and (13b), (22) can be written in the time domain by (24), as shown at the bottom of the page, with $X_{k,\eta,i}^{(t)}$ defined by (17). In order to derive an efficient implementation, (24) can be rewritten after removing columns containing only zeros and by utilizing (19), as given by (25), shown at the bottom of the previous page.

Such system can be split into two parts as

$$\begin{bmatrix} u_0[m] \\ u_1[m+1] \end{bmatrix} = a_R \begin{bmatrix} x_0[m-2R] \\ -x_1[m+1] \end{bmatrix} + \begin{bmatrix} \tilde{u}_0[m] \\ \tilde{u}_1[m+1] \end{bmatrix} \quad (26a)$$

with

$$\begin{bmatrix} \tilde{u}_0[m] \\ \tilde{u}_1[m+1] \end{bmatrix} = \begin{bmatrix} a_0 & -a_1 & a_2 & \cdots & -a_{R-1} & -b_{R-1} & b_{R-2} & \cdots & -b_1 & b_0 \\ b_0 & -b_1 & b_2 & \cdots & -b_{R-1} & a_{R-1} & -a_{R-2} & \cdots & a_1 & -a_0 \end{bmatrix} \times \begin{bmatrix} \mathbf{x}_{0,2,R-1}^{(0)}[m] \\ \mathbf{x}_{1,2,R}^{(1)}[m] \end{bmatrix}. \quad (26b)$$

The first part can be implemented straightforwardly (using only two multiplications), whereas the second part, given by (26b), can be implemented by following the approach used for Type 1, i.e., as shown by (27a) at the bottom of the next page, with

$$\begin{aligned} e_0 &= a_0 + b_0 \\ e_1 &= -a_1 - b_1, \dots, e_{R-1} = -a_{R-1} - b_{R-1} \end{aligned} \quad (27b)$$

$$\begin{aligned} f_0 &= a_0 + b_0 \\ f_1 &= -a_1 - b_1, \dots, f_{R-1} = -a_{R-1} - b_{R-1} \end{aligned} \quad (27c)$$

$\mathbf{x}_{k,\eta,l}^{(t)}$ defined by (17), and

$$\mathbf{x}_{k_0,\eta_0,l_0}^{k_1,\eta_1,l_1,+}[m] = \begin{bmatrix} x_0[m-k_0] + x_1[m-k_1-l_1\eta_1] \\ x_0[m-k_0-\eta_0] + x_1[m-k_1-(l_1-1)\eta_1] \\ \vdots \\ x_0[m-k_0-l_0\eta_0] + x_1[m-k_1] \end{bmatrix}. \quad (27d)$$

$$\begin{bmatrix} u_0[m] \\ u_1[m+1] \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} e_0 & e_1 & \cdots & e_R & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -b_0 & b_1 & \cdots & -b_{R-1} & b_R & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & f_R & \cdots & f_1 & f_0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{0,2,R}^{(0)}[m] \\ \mathbf{x}_{0,2,R}^{(1)}[m] \\ \mathbf{x}_{0,2,R}^{(1)}[m] \end{bmatrix} \quad (20a)$$

$$\begin{bmatrix} u_0[m] \\ u_1[m] \end{bmatrix} = \begin{bmatrix} a_0 & 0 & -a_1 & 0 & \cdots & 0 & a_R & 0 & -b_{R-1} & 0 & \cdots & b_0 & 0 \\ 0 & b_0 & 0 & -b_1 & \cdots & -b_{R-1} & 0 & -a_R & 0 & a_{R-1} & \cdots & 0 & -a_0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{0,1,2R}^{(0)} \\ \mathbf{x}_{0,1,2R}^{(1)} \end{bmatrix} \quad (24)$$

$$\begin{bmatrix} u_0[m] \\ u_1[m+1] \end{bmatrix} = \begin{bmatrix} a_0 & -a_1 & a_2 & \cdots & -a_{R-1} & a_R & 0 & -b_{R-1} & b_{R-2} & \cdots & -b_1 & b_0 \\ b_0 & -b_1 & b_2 & \cdots & -b_{R-1} & 0 & -a_R & a_{R-1} & -a_{R-2} & \cdots & a_1 & -a_0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{0,2,R}^{(0)}[m] \\ \mathbf{x}_{-1,2,R}^{(1)}[m] \end{bmatrix} \quad (25)$$

Fig. 9. Proposed implementation structure for a Type 1 quadruplet for $\mu = 1$ that is part of an FB with $M = 8$ and $N = 33$.

As in the case of Type 1, the coefficients e_k and f_k depend only on the prototype filter coefficients and, as such, can be precalculated.

As an example, the implementation for a Type 2 quadruplet for $M = 8$ and $N = 33$ is considered. As can be seen from (9b), there is one Type 2 quadruplets in this FB, which can be generated for $\mu = 0$ [see also (11a)]. After applying the aforementioned derivation on the quadruplet given by (9b), the following system of equation is obtained:

$$\begin{bmatrix} u_0[m] \\ u_7[m+1] \end{bmatrix} = h_1 \begin{bmatrix} x_1[m-4] \\ -x_0[m+1] \end{bmatrix} + \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} e_0 & e_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_8 & -h_9 & 0 & 0 \\ 0 & 0 & 0 & 0 & f_1 & f_0 \end{bmatrix} \times \begin{bmatrix} x_1[m] \\ x_1[m-2] \\ x_1[m] + x_0[m-3] \\ x_1[m-2] + x_0[m-1] \\ x_0[m-1] \\ x_0[m-3] \end{bmatrix} \quad (28a)$$

with

$$e_0 = h_0 + h_8 \quad e_1 = -h_{16} - h_9 \quad (28b)$$

$$f_0 = h_0 - h_8 \quad f_1 = -h_{16} + h_9. \quad (28c)$$

The implementation is shown in Fig. 10.

V. COMPLEXITY EVALUATION

The complexity of the proposed structures will be evaluated through the number of multiplications and number of additions per M input samples denoted as C^* and C^+ , respectively. First, the complexities for the Type 1 and Type 2 structures are given, and then, a generalization is presented for any combination of M and N . In the complexity evaluation, the given formulas take into account the prototype filter and matrices $\mathbf{S}_{\Delta}^{(2M)}$ and $\mathbf{T}$, as shown in Fig. 3. The implementation of DCT-IV is omitted from the evaluation as it is the same for all cases under consideration.

Fig. 10. Proposed implementation structure for one Type 2 quadruplet that is part of an FB with $M = 8$ and $N = 33$.

For a Type 1 quadruplet, the implementation complexity can be evaluated by

$$C_{T1}^*(R) = 3R + 3 \quad (29a)$$

$$C_{T1}^+(R) = 4R + 5. \quad (29b)$$

As discussed in Section IV-A, R is equal to $K_E - 1$, K_E , or $K_E + 1$ for the structure shown in Fig. 7(a), 7(c), or 7(e), respectively.

For a Type 2 quadruplet, the implementation complexity can be evaluated by

$$C_{T2}^*(R) = 3R + 2 \quad (30a)$$

$$C_{T2}^+(R) = 4R + 3. \quad (30b)$$

As discussed in Section IV-B, R is equal to K_E or $K_E + 1$ for the structure shown in Fig. 7(b) or 7(d), respectively.

For an NPR cosine-modulated M -channel FB, with M being even, synthesized by using a linear-phase prototype filter of order N , with N being odd, the implementation complexity can be evaluated by using one of the following:

$$0 \leq \Delta \leq M - 1$$

$$C^x = \left| \Delta - \frac{M}{2} \right| C_{T1}^x \left(K_E - 1 + \left\lfloor \frac{\Delta}{\frac{M}{2}} \right\rfloor \right) + \left(\frac{M}{2} - \left| \Delta - \frac{M}{2} \right| \right) C_{T2}^x(K_E) \quad (31a)$$

$$M \leq \Delta \leq 2M - 1$$

$$C^x = \left| \Delta - \frac{3M}{2} \right| C_{T1}^x \left(K_E + \left\lfloor \frac{\Delta}{\frac{3M}{2}} \right\rfloor \right) + \left(\frac{M}{2} - \left| \Delta - \frac{3M}{2} \right| \right) C_{T2}^x(K_E + 1) \quad (31b)$$

with K_E and Δ being related to the filter order by (3a). In the aforementioned equation, “ x ” stands for “ $*$ ” or “ $+$ ”, i.e., the same equation is used for evaluating the number of required multiplications and additions.

$$\begin{bmatrix} \tilde{u}_0[m] \\ \tilde{u}_1[m+1] \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} e_0 & e_1 & \cdots & e_{R-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & b_0 & -b_1 & \cdots & b_{R-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & f_{R-1} & \cdots & f_1 & f_0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{0,2,R-1}^{(0)}[m] \\ \mathbf{x}_{0,2,R-1}^{1,2,R-1,+}[m] \\ \mathbf{x}_{1,2,R-1}^{(1)}[m] \end{bmatrix} \quad (27a)$$

Fig. 11. Stopband attenuations for 32-channel NPR cosine-modulated FBs with $\delta_a = \delta_d = 0.01$.

For prototype filters with even order N , the aforementioned expression can be used to get a close-enough estimation by evaluating them for order $N + 1$. FBs with an odd number of channels are not of interest in practice due to the fact that an FB with an even number of channels, having one channel more than the FB with an odd number of channels, in most cases, can be implemented more efficiently than the one with an odd number of channels. Therefore, the complexity estimation for M odd can be evaluated by using the aforementioned equations for $M + 1$ instead of M .

VI. EXAMPLES

This section shows, by means of an example, the benefits of the proposed implementation method. The section is divided into two parts. First, the proposed implementation is compared with the polyphase implementation, and second, it is elaborated why having efficient implementation methods for NPR FBs with an arbitrary prototype filter order is beneficial. It should be pointed out that the complexity evaluation is performed only for the implementation of the prototype filter and matrices $\mathbf{S}_{\Delta}^{(2M)}$ and $\mathbf{T}$. In both cases, the proposed one and the polyphase one, in addition to the prototype filter, the DCT-IV has to be implemented. It is omitted from the following comparison because the implementation cost of the DCT-IV is equal for both approaches.

In order to compare the proposed method with the polyphase one, a family of FBs has been designed, having the following properties: $M = 32$, maximum allowable aliasing and amplitude distortions $\delta_a = \delta_d = 0.01$, stopband edge $\omega_s = \pi/M$, and $N = 95, 97, \dots, 319$. The FBs have been designed by using the method presented in [6], i.e., the stopband attenuation of the prototype filter is minimized subject to the given FB constraints. For these FBs, the achieved stopband attenuations are shown in Fig. 11. It can be noticed that the attenuation is increasing in a monotone continuous manner when the filter order is increasing.⁸ This is not the case for PR FBs where designs other than $N = 2KM - 1$ result in FBs with poor performance [14].

The implementation complexities by using the proposed implementation and the polyphase one are shown in Fig. 12 with the relative comparison shown in Fig. 13. These figures show that the number of required multiplications for the proposed implementation is always lower than that of the polyphase one.

⁸A similar behavior was also observed for different values of M .

Fig. 12. Implementation complexity for FBs shown in Fig. 11 when using $(\dots \times \dots)$ the proposed method and $(\dots + \dots)$ the polyphase method. (a) Multiplication complexity. (b) Addition complexity.

The number of additions in the proposed method is, by $M/2$ addition, higher than that of the polyphase one, independently of the filter order. This is not a problem as the difference is small, particularly for filters of high order, compared with the overall number of addition. Moreover, adders are less costly to implement than multipliers. The numbers of required multiplications and additions in the polyphase case have been evaluated by

$$C_p^* = N + 1 \quad (32a)$$

$$C_p^+ = N + 1. \quad (32b)$$

As already mentioned before, (32b) also includes the $2M$ additions required for implementing matrices $\mathbf{S}_{\Delta}^{(2M)}$ and $\mathbf{T}$, as shown in Fig. 3. In order to make the comparison fairer, it was assumed that both implementation structures, the polyphase one and the proposed one, are implemented according to Fig. 3. The only difference is that the proposed one goes one step further and utilizes the coefficient symmetries, as described in this paper.

By using the proposed implementation method for NPR FBs, systems with lower delays/complexity for given FB requirements can be achieved. This is important due to the fact that, for many applications, the delay introduced by the system is limited by standards. As an example, in Table IV, the numerical data of some characteristic designs is given. As seen from the table, if a design with filters having a 50-dB stopband attenuation is desired with parameters δ_a , δ_d , and ω_s , as given above, then a filter with order $N = 219$ is required. In this case, 22% less multiplications are required by the proposed implementation compared with the polyphase one. In order to compare with the technique proposed in [5], where the efficient implementation applies only to filters with $N = 2KM - 1$, the two nearest $N = 2KM - 1$ cases are given in the table. As it can be seen, the order $N = 191$

Fig. 13. Relative comparison between the proposed and polyphase methods. (a) Multiplication complexity. (b) Addition complexity.

TABLE IV
ATTENUATION AND IMPLEMENTATION COMPLEXITY FOR FBs WITH $M = 32$
AND VARIOUS FILTER ORDERS IN NUMBER OF MULTIPLICATIONS (C^*) AND
ADDITIONS (C^+) PER OUTPUT SAMPLE

N	Att [dB]	Proposed		Method in [5]		Polyphase		Comparison	
		C^*	C^+	$C^*_{[5]}$	$C^+_{[5]}$	C^*_p	C^+_p	C^*/C^*_p	C^+/C^+_p
2.3:32-1=191	47.6	144	208	144	108	192	192	0.75	1.08
219	50.2	172	236	—	—	220	220	0.78	1.07
2.4:32-1=255	58.1	192	272	192	272	256	256	0.75	1.06

has an attenuation lower than 50 dB, and as such, it does not satisfy the requirements of this example. In order to meet the given specification, the technique proposed in [5] has to use an overdesigned filter with order $N = 255$. However, such system has a delay that is 36 samples longer and a more complex implementation compared to the optimum $N = 219$ case. Therefore, the proposed method gives more flexibility when selecting an appropriate NPR FB for a problem at hand.

VII. CONCLUDING REMARKS

In this paper, an implementation method that reduces the number of required multiplications when implementing a linear-phase prototype filter of an arbitrary order used for building an NPR cosine-modulated FB has been proposed. The following remarks regarding the discussion presented in this paper should be made.

First, the method is useful only for NPR FBs. PR FBs can be implemented more efficiently by using a lattice structure. However, the PR FBs are restricted to orders $N = 2KM - 1$, whereas the proposed method can be used with an arbitrary filter order.

Second, although this paper mainly concentrates on odd filter orders and even number of channels, the method can also be used to implement prototype filters of even order, as well as FBs with an odd number of channels. In these cases, in addition to Type 1 and Type 2 quadruplets, some trivial relations have to be implemented.

Third, it is shown how to efficiently implement the cosine-modulated part of the FB for an even number of channels and odd filter orders. This can be extended to other numbers of channels and filter orders. The paper concentrated on the aforementioned cases as those are the most useful ones from a practical viewpoint.

Fourth, the proposed implementation does not depend on the properties of the FB or how the FB has been designed. Therefore, the proposed method can be used for any existing or newly designed FB having a linear-phase prototype filter.

Fifth, as shown in the example section, in some cases, FBs with lower delays can be used without increasing the number of required multiplications. This is very important in many applications.

Sixth, in this paper, analysis-synthesis systems have been considered. However, the proposed implementation method can also be used for synthesis-analysis systems, also known as transmultiplexers.

APPENDIX A

This appendix shows that the modulation part of a cosine-modulated FB with a linear-phase prototype filter of an odd order having an even number of channels can always be implemented by using a fast DCT.⁹ Although, this appendix concentrates on the aforementioned cases, a similar principle can be also applied for other filter orders and/or numbers of channels.

It has been shown in [1] that, for FBs with even numbers of channels and filter orders equal to $N = 2K_E M - 1$, with K_E being even, a cosine-modulation matrix given by (5a) can be implemented by using a DCT-IV transform as given by (6a)–(6h)¹⁰ (DCT-IV can be implemented by a fast DCT [11], [12]). In this special case, the cross-connection matrix $\mathbf{S}_{\Delta}^{(2M)}$ becomes an identity matrix because $\Delta = 0$. This special order will be denoted here as N_K , i.e., $N_K = 2K_E M - 1$. The corresponding modulation matrix denoted as $[\mathbf{C}_{NK}]_{k,l}$ is given by

$$[\mathbf{C}_{NK}]_{k,l} = 2 \cos \left(\frac{\pi}{M} \left(k + \frac{1}{2} \right) \left(l - \frac{N_K}{2} \right) + \theta_k \right) \quad (33a)$$

for $k = 0, 1, \dots, M - 1$ and $l = 0, 1, \dots, 2M - 1$, and θ_k is

$$\theta_k = (-1)^k \frac{\pi}{4}. \quad (33b)$$

The goal now is to show that the cosine-modulation matrix $[\mathbf{C}_1]_{k,l}$ of an FB with filter order $N = 2K_E M + 2\Delta - 1$ can be implemented as

$$[\mathbf{C}_1]_{k,l} = [\mathbf{C}_{NK}]_{k,l} \mathbf{S}_{\Delta}^{(2M)} \quad (34)$$

with $\mathbf{S}_{\Delta}^{(2M)}$ given by (6h).

⁹A similar proof for a different type of cosine-modulation functions than the one used in this paper can be found in [18].

¹⁰Similar relations can be derived for $N = 2K_O M - 1$, with K_O being an odd integer. However, it has turned out that, for the discussion in this paper, it is beneficial to use only the relations for even K_E 's.

In order to prove this, first, the implementation for $N = N_K + 2$ is considered. For this case, the modulation matrix is given by

$$[\mathbf{C}_1]_{k,l} = 2 \cos \left(\frac{\pi}{M} \left(k + \frac{1}{2} \right) \left(l - \frac{N_K + 2}{2} \right) + \theta^k \right) \quad (35)$$

for $k = 0, 1, \dots, M-1$ and $l = 0, 1, \dots, 2M-1$. By comparing (33a) and (35), it can be observed that

$$l - \frac{N_K + 2}{2} = l - \frac{N_K}{2} - 1 = (l-1) - \frac{N_K}{2} \quad (36a)$$

i.e., matrices $[\mathbf{C}_1]_{k,l}$ and $[\mathbf{C}_{NK}]_{k,l}$ have identical columns

$$[\mathbf{C}_1]_{k,l+1} = [\mathbf{C}_{NK}]_{k,l} \quad (36b)$$

for $k = 0, 1, \dots, M-1$ and $l = 0, 1, \dots, 2M-2$. The only different columns are the first column in matrix $[\mathbf{C}_1]_{k,l}$ given as

$$\begin{aligned} [\mathbf{C}_1]_{k,0} &= 2 \cos \left(\frac{\pi}{M} \left(k + \frac{1}{2} \right) \left(0 - \frac{N_K + 2}{2} \right) + \theta^k \right) \\ &= 2 \cos \left(\frac{\pi}{M} \left(k + \frac{1}{2} \right) \left(-K_E M - \frac{1}{2} \right) + \theta^k \right) \end{aligned} \quad (37a)$$

and the last column in matrix $[\mathbf{C}_{NK}]_{k,l}$ given as

$$\begin{aligned} [\mathbf{C}_{NK}]_{k,2M-1} &= 2 \cos \left(\frac{\pi}{M} \left(k + \frac{1}{2} \right) \left(2M-1 - \frac{N_K}{2} \right) + \theta^k \right) \\ &= 2 \cos \left((2k+1)\pi + \frac{\pi}{M} \left(k + \frac{1}{2} \right) \right. \\ &\quad \times \left. \left(-K_E M - \frac{1}{2} \right) + \theta^k \right). \end{aligned} \quad (37b)$$

By applying the cosine transformation $\cos(\alpha + \beta) = \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta)$ on (37b) and noticing that

$$\cos((2k+1)\pi) = -1 \quad (38a)$$

$$\sin((2k+1)\pi) = 0 \quad (38b)$$

(37b) becomes

$$[\mathbf{C}_{NK}]_{k,2M-1} = -2 \cos \left(\frac{\pi}{M} \left(k + \frac{1}{2} \right) \left(-K_E M - \frac{1}{2} \right) + \theta^k \right) \quad (39)$$

that is,

$$[\mathbf{C}_{NK}]_{k,2M-1} = -[\mathbf{C}_1]_{k,0}. \quad (40)$$

This means that, for $N = N_K + 2$, all columns of matrix $[\mathbf{C}_1]_{k,l}$ are shifted versions of matrix $[\mathbf{C}_{NK}]_{k,l}$ with one of the columns having a different sign. Therefore, by simply rearranging the input signals going into the cosine-modulation matrix, the matrix $[\mathbf{C}_{NK}]_{k,l}$ can be used for both of those cases. The required rearrangement can be expressed by the following matrix (see also Fig. 3):

$$\mathbf{S}_1^{(2M)} = \begin{bmatrix} \mathbf{0}_{2M-1,1} & \mathbf{I}_{2M-\Delta} \\ -1 & \mathbf{0}_{1,2M-1} \end{bmatrix} \quad (41a)$$

that is,

$$\mathbf{C}_1 = \mathbf{C}_{NK} \mathbf{S}_1^{(2M)}. \quad (41b)$$

It can be easily shown that the same principle can be applied for other filter orders up to $N = N_K + 4M - 2$. For example, for $N = N_K + 4$

$$[\mathbf{C}_1]_{k,l+2} = [\mathbf{C}_{NK}]_{k,l} \quad (42a)$$

for $k = 0, 1, \dots, M-1$ and $l = 0, 1, \dots, 2M-3$, and

$$[\mathbf{C}_{NK}]_{k,2M-2} = -[\mathbf{C}_1]_{k,0} \quad (42b)$$

$$[\mathbf{C}_{NK}]_{k,2M-1} = -[\mathbf{C}_1]_{k,1}. \quad (42c)$$

Consequently

$$\mathbf{S}_2^{(2M)} = \begin{bmatrix} \mathbf{0}_{2M-2,2} & \mathbf{I}_{2M-2} \\ -\mathbf{I}_2 & \mathbf{0}_{2,2M-2} \end{bmatrix} \quad (42d)$$

$$\mathbf{C}_1 = \mathbf{C}_{NK} \mathbf{S}_2^{(2M)}. \quad (42e)$$

In the more general case, for $N = 2K_E M - 1 + 2\Delta$, the cross-connection matrix $\mathbf{S}_\Delta^{(2M)}$ becomes as given by (6h), thereby proving (34).

Finally, it should be pointed out that, for $N = N_K + 4M$, $[\mathbf{C}_1]_{k,l} = -[\mathbf{C}_{NK}]_{k,l}$ for $k = 0, 1, \dots, M-1$ and $l = 0, 1, \dots, 2M-1$. Because all columns change sign, this can be implemented by changing the sign of the input or output signals. In the implementation, this is performed by the scaling factor λ . Therefore, for $N = N_K + 4M$

$$[\mathbf{C}_1]_{k,l} = [\mathbf{C}_{NK}]_{k,l} \quad (43a)$$

for $k = 0, 1, \dots, M-1$ and $l = 0, 1, \dots, 2M-1$ with

$$\lambda = -\lambda_{NK} \quad (43b)$$

and λ_{NK} corresponding to the $N = N_K$ case.

Here, only the analysis modulation matrix is considered. The same principle can be applied for deriving an implementation for the synthesis modulation matrix.

APPENDIX B

This appendix shows that the symmetry relations given by (13a) and (13b) hold for the quadruplet shown in Fig. 7(a), i.e., (9a). This corresponds to the Type 1 quadruplet shown in Fig. 8(a). For the quadruplets shown in Fig. 7(b)–7(e), i.e., (9b)–(9e), a proof similar to the one here can be done.

According to Fig. 8(a), Fig. 7(a), and (9a), it follows that

$$A(z) = G_{\mu+\Delta}(z) \quad (44a)$$

$$B(z) = G_{\mu+\Delta+M}(z) \quad (44b)$$

$$C(z) = G_{M-\mu+\Delta-1}(z) \quad (44c)$$

$$D(z) = G_{2M-\mu+\Delta-1}(z) \quad (44d)$$

for $0 \leq \Delta \leq M/2 - 1$ and $\mu = \Delta, \Delta + 1, \dots, M/2 - 1$ with Δ given by (3c). Combining the aforementioned relations with (13a) and (13b), it turns out that, for the case under consideration, the following relations have to be proven:

$$g_{\mu+\Delta}[n] = g_{2M-\mu+\Delta-1}[R_0 - n], \quad \text{for } n = 0, 1, \dots, R_0 \quad (45a)$$

$$g_{M-\mu+\Delta-1}[n] = g_{\mu+\Delta+M}[R_1 - n], \quad \text{for } n = 0, 1, \dots, R_1. \quad (45b)$$

According to (4c), the orders of the polyphase components can be derived as

$$N_\xi = N_A = \left\lfloor \frac{N+1}{2M} \right\rfloor - 1, \quad \text{for } \xi = 0, 1, \dots, P-1 \quad (46a)$$

$$N_\xi = N_B = \left\lfloor \frac{N+1}{2M} \right\rfloor - 1, \quad \text{for } \xi = P, P+1, \dots, 2M-1 \quad (46b)$$

where

$$P = N + 1 - 2M \left\lfloor \frac{N+1}{2M} \right\rfloor. \quad (46c)$$

This means that the first P polyphase filters are of order N_A and the other $2M - P$ polyphase filters are of order N_B . By combining (3a) with (46c), it turns out that

$$P = 2K_E M + 2\Delta - 2M \left\lfloor K_E + \frac{\Delta}{2M} \right\rfloor = 2\Delta. \quad (47)$$

The last equality in (47) holds because $\Delta < M/2$.

As discussed in [15]–[17], the polyphase filter coefficients present mirror image symmetries for polyphase filter pairs. More specifically

$$g_\zeta[n] = g_{P-1-\zeta}[N_A - n], \quad \text{for } \zeta = 0, 1, \dots, P-1 \quad (48a)$$

$$g_\zeta[n] = g_{2M-1+P-\zeta}[N_B - n], \quad \text{for } \zeta = P, P+1, \dots, 2M-1. \quad (48b)$$

For the case under consideration, $\mu \geq \Delta$. Therefore, $\mu + \Delta \geq 2\Delta = P$ and $R_0 = N_B = N_{KE} - 1$. Consequently, (48b) shows that the $(\mu + \Delta)$ -th polyphase filter is the reverse of the $(2M - 1 + P - (\mu + \Delta))$ -th polyphase filter. Since

$$\begin{aligned} 2M - 1 + P - (\mu + \Delta) &= 2M - 1 + 2\Delta - \mu - \Delta \\ &= 2M - 1 - \mu + \Delta. \end{aligned} \quad (49)$$

Equation (45a) is proven. Equation (45b) can be proven in exactly the same way, since $\mu + \Delta + M \geq 2\Delta + M \geq P$.

REFERENCES

- [1] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
- [2] N. J. Fliege, *Multirate Digital Signal Processing*. Chichester, U.K.: Wiley, 1994.
- [3] H. S. Malvar, *Signal Processing With Lapped Transforms*. Norwood, MA: Artech House, 1992.
- [4] R. Bregović and T. Saramäki, "An efficient approach for designing nearly perfect-reconstruction low-delay cosine-modulated filter banks," in *Proc. IEEE Int. Symp. Circuits Syst.*, Scottsdale, AZ, May 2002, vol. 1, pp. 825–828.
- [5] J. Alhava and A. Viholainen, "Coefficient quantization in nearly perfect-reconstruction cosine-modulated filter banks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Istanbul, Turkey, Jun. 2000, pp. 536–539.
- [6] T. Saramäki and R. Bregović, "An efficient approach for designing nearly perfect-reconstruction cosine-modulated and modified DFT filter banks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Salt Lake City, UT, May 2001, vol. 6, pp. 3617–3620.
- [7] W.-S. Lu, T. Saramäki, and R. Bregović, "Design of practically perfect-reconstruction cosine-modulated filter banks: A second-order cone programming approach," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 3, pp. 552–563, Mar. 2004.
- [8] P. N. Heller, T. Karp, and T. Q. Nguyen, "A general formulation of modulated filter banks," *IEEE Trans. Signal Process.*, vol. 47, no. 4, pp. 986–1002, Apr. 1999.
- [9] T. Karp, A. Mertins, and G. Schuller, "Efficient biorthogonal cosine-modulated filter banks," *Signal Process.*, vol. 81, no. 5, pp. 997–1016, May 2001.
- [10] F. Cruz-Roldán and M. Monteagudo-Prim, "Efficient implementation of nearly perfect reconstruction FIR cosine-modulated filterbanks," *IEEE Trans. Signal Process.*, vol. 52, no. 9, pp. 2661–2664, Sep. 2004.
- [11] K. R. Rao and P. Yip, *Discrete Cosine Transform—Algorithms, Advantages, Applications*. New York: Academic, 1990.
- [12] C. W. Kok, "Fast algorithm for computing discrete cosine transform," *IEEE Trans. Signal Process.*, vol. 45, no. 3, pp. 757–760, Mar. 1997.
- [13] K. K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*. New York: Wiley, 1999.
- [14] T. Q. Nguyen and R. D. Koilpillai, "The theory and design of arbitrary-length cosine-modulated filter banks and wavelets, satisfying perfect reconstruction," *IEEE Trans. Signal Process.*, vol. 44, no. 3, pp. 473–483, Mar. 1996.
- [15] Z. J. Mou, "Symmetry exploitation in digital interpolators/decimators," *IEEE Trans. Signal Process.*, vol. 44, no. 10, pp. 2611–2615, Oct. 1996.
- [16] Y. J. Yu, Y. C. Lim, and T. Saramäki, "Restoring coefficient symmetry in polyphase implementation of linear-phase filters," *Circuits Syst. Signal Process.*, vol. 25, no. 2, pp. 253–264, May 2006.
- [17] R. Bregović, T. Saramäki, Y. J. Yu, and Y. C. Lim, "An efficient implementation of linear-phase FIR filters for a rational sampling rate conversion," in *Proc. IEEE Int. Symp. Circuits Syst.*, Island of Kos, Greece, May 2006, pp. 5395–5398.
- [18] G. D. T. Schuller and T. Karp, "Modulated filter banks with arbitrary system delay: Efficient implementations and the time-varying case," *IEEE Trans. Signal Process.*, vol. 48, no. 3, pp. 737–748, Mar. 2000.

Robert Bregović (M'97) was born in Varaždin, Croatia, in 1970. He received the Dipl.Ing. and M.Sc. degrees in electrical engineering from the Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia, in 1994 and 1998, respectively, and the D.Sc.(Tech.) degree (with honors) in information technology from the Tampere University of Technology (TUT), Tampere, Finland, in 2003.

From 1994 to 1998, he was an Assistant with the Department of Electronic Systems and Information Processing, Faculty of Electrical Engineering and Computing, University of Zagreb. In 1999, he was a Visiting Researcher with the Tampere International Center for Signal Processing, Institute of Signal Processing Laboratory, Department of Information Technology, TUT, where he was a Researcher from January 2000 to August 2003 and has been a Postdoctoral Researcher since September 2003. His research interests are in digital signal processing, particularly in multirate signal processing, digital filterbanks, and generating optimization procedures for designing digital filters and filterbanks for various applications, as well as optimizing DSP algorithms for digital signal processor implementations.

Ya Jun Yu (S'99–M'05–SM'09) received the B.Sc. and M.Eng. degrees in biomedical engineering from Zhejiang University, Hangzhou, China, in 1994 and 1997, respectively, and the Ph.D. degree in electrical and computer engineering from the National University of Singapore, Singapore, in 2004.

Since 2005, she has been with the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore, where she is currently an Assistant Professor. From 1997 to 1998, she was a Teaching Assistant with Zhejiang University. She joined the Department of Electrical and Computer Engineering, National University of Singapore, as a Post Master Fellow in 1998 and remained in the same department as a Research Engineer until 2004. In 2002, she was a Visiting Researcher with the Tampere University of Technology, Tampere, Finland, and the Hong Kong Polytechnic University, Kowloon, Hong Kong. She joined the Temasek Laboratories, NTU, Singapore, as a Research Fellow in 2004. Her research interests include digital signal processing and VLSI circuit and system design.

Ari Viholainen was born in Nokia, Finland, on May 26, 1972. He received the M.Sc. (with distinction) and D.Tech. degrees in information technology from the Tampere University of Technology (TUT), Tampere, Finland, in 1998 and 2004, respectively.

Currently, he is a Senior Researcher with the Department of Communications Engineering, TUT. His research interests are in digital signal processing and digital communications, particularly in multirate filter banks and multicarrier systems.

Yong Ching Lim (S'79–M'82–SM'92–F'00) received the A.C.G.I. and B.Sc. degrees and the D.I.C. and Ph.D. degrees in electrical engineering from Imperial College, University of London, London, U.K., in 1977 and 1980, respectively.

Since 2003, he has been with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore, where he is currently a Professor. From 1980 to 1982, he was a National Research Council Research Associate with the Naval Postgraduate School, Monterey, CA. From

1982 to 2003, he was with the Department of Electrical Engineering, National

University of Singapore. His research interests include digital signal processing and VLSI circuit and system design.

Dr. Lim was a recipient of the 1996 IEEE Circuits and Systems Society's Guillemin–Cauer Award, the 1990 IREE (Australia) Norman Hayes Memorial Award, 1977 IEE (U.K.) Prize, and the 1974–1977 Siemens Memorial (Imperial College) Award. He was a Lecturer for the IEEE Circuits and Systems Society under the distinguished lecturer program from 2001 to 2002 and an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS from 1991 to 1993 and from 1999 to 2001. He also served as an Associate Editor for *Circuits, Systems and Signal Processing* from 1993 to 2000. He was the Chairman of the DSP Technical Committee of the IEEE Circuits and Systems Society from 1998 to 2000. He served in the Technical Program Committee's DSP Track as the Chairman for the IEEE International Symposium on Circuits and Systems 1997 (ISCAS'97) and IEEE ISCAS'00 and as a Cochairman for the IEEE ISCAS'99. He was the General Chairman for IEEE Asia Pacific Conference on Circuits and Systems (APCCAS) 2006 and is a General Cochair for the IEEE International Symposium on Circuits and Systems (ISCAS) 2009.