# EFFICIENTLY VLSI-REALIZABLE PROTOTYPE FILTERS FOR MODULATED FILTER BANKS


*Tanja Karp*
Hamburg Univ. of Technology
Telecommunications Institute
21071 Hamburg, Germany
karp@tu-harburg.d400.de

*Alfred Mertins*
University of Kiel
Kaiserstr. 2
24143 Kiel, Germany
am@techfak.uni-kiel.de

*Truong Q. Nguyen*
Univ. of Wisconsin, ECE Dept.
1415 Engr. Dr
Madison, WI 53706, USA
nguyen@ece.wisc.edu



## ABSTRACT

This paper presents methods for the efficient realization of prototype filters for modulated filter banks. The implementation is based on the lattice structure of the polyphase filters. The lattice coefficients, representing rotations, are approximated by a small number of simple $\mu$-rotations each of which can be realized by some shift and add operations instead of a multiplication. Since the lattice structure is robust against coefficient quantization we do not loose the perfect reconstruction (PR) property of the filter bank when doing this approximation. Frequency responses of the original and approximated prototype filters are compared in terms of complexity and stopband attenuation.


## 1. INTRODUCTION

Modulated filter banks are known to provide a very efficient realization. They consist of two main stages: the polyphase filters of the prototype and a discrete cosine or Fourier transform. Using fast algorithms for the transform (i.e. fast DCT or FFT) the main computational complexity remains with the multiplications during the filter operation [1].

Although many filter design methods for PR cosine-modulated filter banks have been developed within the last years, see e.g. [2, 3], none of these methods takes into consideration the implementation cost due to the wordlengths of the filter coefficients.

In this paper we show how to approximate the rotation angles in the lattice structure by some consecutive $\mu$-rotations which can be realized by a small number of shift and add operations. Consequently, prototype filters can be efficiently implemented in VLSI using a structure that is not only extremely regular but also designated for a parallel implementation.

Figure 1 shows the analysis part of an $M$-channel cosine-modulated filter bank as derived in [2]. Herein $G_k(z)$, $k = 0, \ldots, 2M-1$, denote the type-1 polyphase components of the prototype filter.

Each block of two polyphase filters in Figure 1 can be realized as a lattice. Figure 2 shows the well-known lattice structure [2] of one block in Figure 1.

Typically, prototype filters are designed such as to have linear phase. In this case two polyphase filters have the same coefficients but in reverse order. For a linear-phase

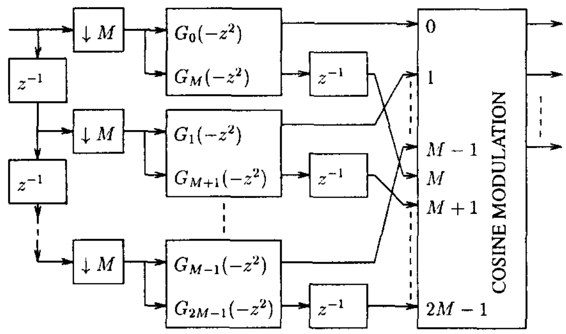


Figure 1: Cosine-modulated $M$-channel analysis filter bank

prototype of length $N = 2rM$ the polyphase filters $G_k(z)$ and $G_{2M-1-k}(z)$ are related by:

$$g_k(m) = g_{2M-1-k}(r-1-m), \qquad m = 0, \ldots, r-1.$$

The lattice structure for the filters $G_{M-1-k}(z)$ and $G_{2M-1-k}(z)$, respectively, contains the same rotation angles $\theta_{k,l}$, and therefore the same values $s_{k,l}$ and $c_{k,l}$, as the one for $G_k(z)$ and $G_{M+k}(z)$. It is depicted in Figure 3.

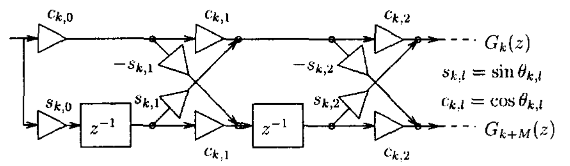


Figure 2: Lattice structure of polyphase filters $G_k(z)$ and $G_{k+M}(z)$

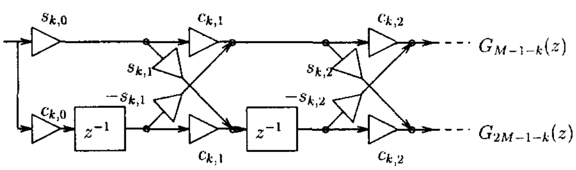


Figure 3: Lattice structure of polyphase filters $G_{M-1-k}(z)$ and $G_{2M-1-k}(z)$

One common method to realize rotations is the CORDIC algorithm [4]. A rotation by an angle $\theta$ is realized as a cascade of $w+1$ so called $\mu$-rotations ($w$ being the coefficient wordlength). Each $\mu$-rotation consists of two shift and add operations plus a scaling factor.

For $0 \leq |\theta| < \pi/4$ the rotation matrix $\mathbf{W}$ can be described as

$$\mathbf{W} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} = \prod_{v=0}^{w} r_v \begin{bmatrix} 1 & \sigma_v 2^{-v} \\ -\sigma_v 2^{-v} & 1 \end{bmatrix} \quad (1)$$

with $r_v = \frac{1}{\sqrt{1+\sigma_v^2 2^{-2v}}}$ and $\sigma_v \in \{-1, +1\}$, whereas angles in the range $\pi/4 < \pm\theta < \pi/2$ are realized as

$$\mathbf{W} = \begin{bmatrix} 0 & \pm 1 \\ \mp 1 & 0 \end{bmatrix} \begin{bmatrix} \cos(\pi/2 \mp \theta) & \sin(\pi/2 \mp \theta) \\ -\sin(\pi/2 \mp \theta) & \cos(\pi/2 \mp \theta) \end{bmatrix} \quad (2)$$

where the first matrix describes a rotation by $\pm\pi/2$.

## 2. APPROXIMATION OF ROTATION ANGLES

The idea presented in [5] for an efficient implementation of orthogonal wavelets was to perform only a limited number of CORDIC $\mu$-rotations, i.e. choosing $\sigma_v \in \{-1, 0, +1\}$ in (1) or (2) with most values equal zero, yielding an approximation of the original angles.

The angles $\theta_i = \pm\arctan(2^{-i})$ are realizable with a single $\mu$-rotation of the CORDIC algorithm. A finer grid of angles can be obtained either by combinations of $\mu$-rotations or, at the cost of more complex $\mu$-rotations, by adding entries to the $\mu$-rotation matrix $\mathbf{W}_\mu$:

$$\mathbf{W}_\mu = r_\mu \begin{bmatrix} 1 + \sum_v \rho_v 2^{-v} & \sum_v \sigma_v 2^{-v} \\ -\sum_v \sigma_v 2^{-v} & 1 + \sum_v \rho_v 2^{-v} \end{bmatrix}, \quad (3)$$

$$r_\mu = \frac{1}{\sqrt{(1+\sum_v \rho_v 2^{-v})^2 + (\sum_v \sigma_v 2^{-v})^2}},$$

$$\sigma_v, \rho_v \in \{-1, 0, +1\}, \quad v = 0, \dots, w.$$

The definition (3) also includes the so called Class 2 and Class 3 $\mu$-rotations defined in [6]. The lattice realization of $r_\mu \mathbf{W}_\mu$ is depicted in Figure 4.

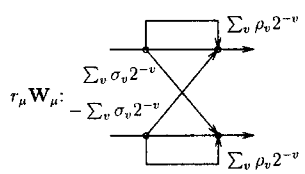


Figure 4: Lattice realization performing $r_\mu \mathbf{W}_\mu$

The implementation cost of the lattice structure depends on the number of $\mu$-rotations needed for a sufficiently fine approximation of the angles, on the number of shift and add operations within each $\mu$-rotation, and on the number of shift and add operations needed for the quantization of a common scaling factor $r_k$ at the entrance of the lattice.

A more or less precise approximation of the rotation angles influences the frequency response of the prototype filter, but it does not affect the PR property of the filter bank. However, quantizing the scaling factor $r_k$ yields non-PR property of the bank and should therefore be done carefully and with respect to the desired accuracy.

## 3. QUANTIZATION OF THE SCALING FACTOR

The scaling factors $r_k$ can be approximated by factorization or as a sum of binaries. Common factors of all $r_k$, $k = 0, \dots, M/2-1$, can be implemented once at the output of the synthesis filter bank. This simply causes a new scaling of all subband signals.

When using quantized scaling factors $\hat{r}_{k,a}$ for the analysis and $\hat{r}_{k,s}$ for the synthesis, we introduce an error $\delta_k$ in the PR condition [7]:

$$G_k(z)\tilde{G}_k(z) + G_{k+M}(z)\tilde{G}_{k+M}(z) = \frac{1}{2M} + \delta_k \quad (4)$$

In order to guarantee PR of the filter bank, we have to ensure that the error $\delta_k$ does not lead to any false decision when rounding the reconstructed signal to the original wordlength.

If we assume the input signal to be integer-valued and in the range of $[-x_{max}, x_{max}-1]$ then $\hat{r}_{k,a}$ and $\hat{r}_{k,s}$ have to be derived from the unquantized scaling factor $r_{k,a} = r_{k,s} = r_k$ such that the following inequality holds true:

$$\max(2M \cdot \delta_k) = x_{max} \left| (\hat{r}_{k,a}\hat{r}_{k,s})/r_k^2 - 1 \right| < 0.5. \quad (5)$$

## 4. EFFICIENT REALIZATION OF PROTOTYPES

Different constraints do arise when implementing short or long prototype filters efficiently. In the case of short prototypes (1 to 2 taps per polyphase filter), a coarse approximation of the angles is mostly sufficient for good frequency responses. The amount of shift and add operations needed for the quantization of the scaling factor is relatively high since there is one scaling factor to be implemented for every block of two polyphase filters consisting of one to two lattice stages. Therefore $\mu$-rotations should be chosen such as to provide a simple scaling factor.

For long filters, the situation nearly becomes the opposite: each lattice block with one scaling factor consists of several lattice stages and a good approximation of the rotation angles is needed in order to obtain the desired prototype frequency response. Therefore the amount of shift and add operations needed for realizing the rotations is much higher than for the scaling factor.

In both cases, we start with a given PR prototype filter and calculate the exact rotation angles first.

### 4.1. Realization of Short Filters

An efficient implementation for short prototype filters can be obtained in the following way:

- In a first step we have to choose the maximal complexity of every lattice stage in terms of additions (neglecting the scaling factor) and the maximal number of shifts ($w$). A reasonable choice is e.g. 6 additions per rotation angle and $w = 10$.
- For each rotation angle we calculate all $\mu$-rotations that approximate the exact angle up to a certain error and that do not exceed the allowed complexity.

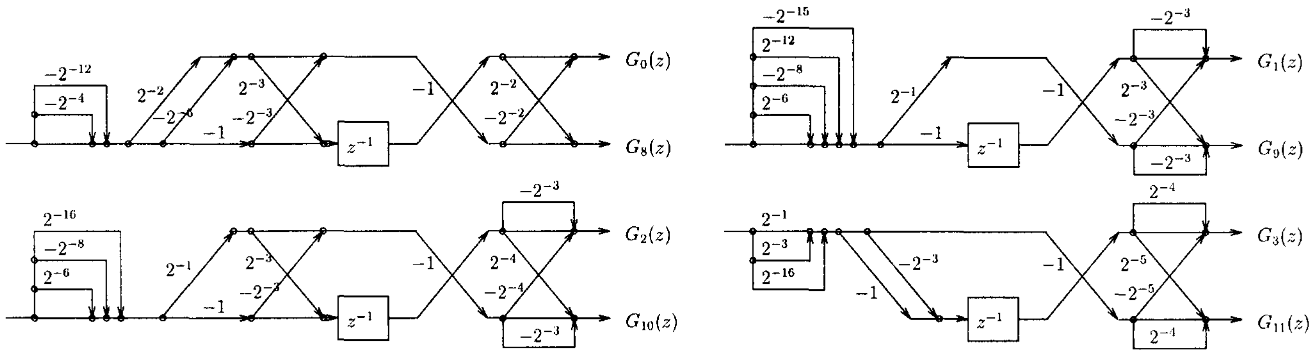


Figure 5: Lattice implementation of a length-32 prototype for an 8-channel cosine-modulated filter bank. The remaining 4 lattice blocks have to be realized according to Figure 3. The integer-valued input signal is in the range $[-2^{15}, 2^{15}-1]$.

- For each lattice filter we calculate the scaling factors for all possible combinations of $\mu$-rotations of the different lattice stages. These scaling factors are quantized with the necessary accuracy according to (5).
- We choose the subset of $\mu$-rotations being realizable with the minimum total number of shift and add operations.
- Out of the remaining realization possibilities we pick the one with the best approximation of the original frequency response.

#### *4.1.1. Example: $N = 32$, $M = 8$*

The above design algorithm is applied to a length-32 linear-phase prototype for an 8-channel cosine-modulated filter bank. The range of the integer-valued input signal is supposed to be $x \in [-2^{15}, 2^{15}-1]$. Figure 5 shows the realization of 8 polyphase filters of the analysis filter bank. Note that the first lattice stage differs from the following ones, see Figure 2. The remaining 8 polyphase filters can be obtained using the structure shown in Figure 3. In the synthesis bank, all scaling factors have to be divided by $2M = 2^{-5}$, which is a pure shift operation. In average, 2 additions and 2.125 shift operations have to be performed per prototype coefficient. The frequency responses of the original prototype designed with the QCLS-algorithm [3] as well as the one realized according to Figure 5 are depicted in Figure 6.

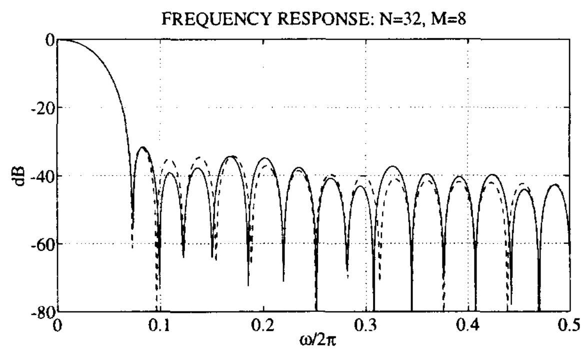


Figure 6: Frequency responses of original prototype (– – –) and efficient realization (——); $N = 32$, $M = 8$.

### 4.2. Realization of Long Filters

The computational complexity of the algorithm proposed for short prototype filters is too high as to be applied to long prototypes. If each lattice block has e.g. 8 lattice stages the number of possible $\mu$-rotations is too large for calculating the scaling factors for all possible combinations and choosing the best one afterwards. We therefore propose the following procedure for long polyphase filters:

- For each rotation angle, we find the best approximation using $\mu$-rotations with a maximal number of shift and add operations. Generally, 8 additions and $w = 12$ should be a choice allowing a grid of $\mu$-rotations being fine enough.
- We iteratively replace the approximation of the least sensitive angle by $\mu$-rotations with less shift and add operations. This pruning is continued as long as this does not significantly change the frequency response of our prototype filter.
- Once we have found a set of $\mu$-rotations with a minimum number of shift and add operations we quantize the scaling factor of each lattice according to (5). If there are several possibilities to realize the same approximated rotation, but with different scaling factors, we choose the one resulting in the least expensive scaling factor.

#### *4.2.1. Example: $N = 128$, $M = 8$*

Figure 7 shows the frequency response of a length-128 prototype for an 8-channel cosine-modulated filter bank designed with the QCLS-algorithm [3] and of the prototype allowing an efficient realization. The non-zero values $\sigma_{i,v}$ and $\rho_{i,v}$ of the $\mu$-rotations in (3) are listed in Table 1. The first index denotes position of the $\mu$-rotation for a fixed angle. The first row describes if the $\mu$-rotations have to be preceded by a $\pm\pi/2$ rotation, which can be performed without any additions or shift operations. The rotation angles have been replaced by coarser approximations as long as the stopband attenuation did not rise above -80 dB. In average 3.16 additions and 3.29 shift operations are performed per prototype coefficient.

Table 1:
Non-zero coefficients and scaling factors of the $\mu$-rotations $\mathbf{W}_\mu$, necessary for the prototype approximation in Figure 7

| $k\backslash l$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | $-\pi/2$<br>$\sigma_{1,2}=+1$<br>$\sigma_{1,12}=-1$<br>$\rho_{1,1}=+1$ | $\sigma_{1,5}=-1$<br>$\sigma_{2,1}=+1$<br>$\sigma_{2,2}=+1$<br>$\sigma_{2,6}=+1$ | $-\pi/2$<br>$\sigma_{1,6}=+1$<br>$\sigma_{1,10}=+1$<br>$\rho_{1,6}=-1$ | $-\pi/2$<br>$\sigma_{1,3}=+1$<br>$\sigma_{1,5}=+1$<br>$\sigma_{2,1}=+1$<br>$\sigma_{2,3}=+1$ | $\pi/2$<br>$\sigma_{1,2}=-1$<br>$\sigma_{2,6}=+1$<br>$\sigma_{3,3}=-1$<br>$\sigma_{3,4}=-1$ | $-\pi/2$<br>$\sigma_{1,7}=+1$<br>$\sigma_{2,2}=+1$<br>$\sigma_{3,4}=-1$ | $-\pi/2$<br>$\sigma_{1,3}=+1$<br>$\rho_{1,4}=-1$ | $\pi/2$<br>$\sigma_{1,2}=+1$<br>$\sigma_{2,2}=+1$<br>$\sigma_{3,0}=-1$<br>$\rho_{3,6}=+1$ |
| 1 | $\pi/2$<br>$\sigma_{1,4}=+1$<br>$\sigma_{2,3}=-1$<br>$\sigma_{2,8}=-1$ | $\pi/2$<br>$\sigma_{1,3}=-1$<br>$\sigma_{1,10}=-1$<br>$\rho_{1,3}=+1$ | $\sigma_{1,0}=-1$<br>$\sigma_{2,2}=+1$ | $-\pi/2$<br>$\sigma_{1,6}=-1$<br>$\sigma_{2,4}=+1$<br>$\rho_{2,6}=-1$ | $\sigma_{1,1}=+1$<br>$\sigma_{1,3}=+1$<br>$\sigma_{1,6}=+1$ | $\pi/2$<br>$\sigma_{1,4}=-1$<br>$\sigma_{1,9}=-1$ | $\pi/2$<br>$\sigma_{1,2}=-1$<br>$\sigma_{2,1}=-1$<br>$\rho_{2,2}=+1$ | $\pi/2$<br>$\sigma_{1,10}=+1$<br>$\sigma_{2,3}=-1$<br>$\sigma_{2,6}=+1$ |
| 2 | $\pi/2$<br>$\sigma_{1,3}=-1$<br>$\sigma_{2,3}=-1$<br>$\sigma_{3,6}=+1$ | $\pi/2$<br>$\sigma_{1,4}=-1$<br>$\sigma_{2,9}=+1$<br>$\rho_{2,3}=+1$ | $\sigma_{1,1}=-1$<br>$\sigma_{2,4}=-1$<br>$\rho_{2,3}=-1$ | $-\pi/2$<br>$\sigma_{1,4}=+1$<br>$\sigma_{1,7}=-1$<br>$\rho_{1,4}=+1$ | $\sigma_{1,4}=-1$<br>$\sigma_{2,3}=+1$<br>$\sigma_{2,6}=+1$ | $\pi/2$<br>$\sigma_{1,3}=-1$<br>$\sigma_{2,7}=+1$<br>$\rho_{2,3}=-1$ | $\pi/2$<br>$\sigma_{1,5}=+1$<br>$\sigma_{2,1}=-1$<br>$\sigma_{2,5}=+1$ | $\pi/2$<br>$\sigma_{1,6}=-1$<br>$\rho_{1,5}=-1$ |
| 3 | $-\pi/2$<br>$\sigma_{1,2}=+1$<br>$\sigma_{2,1}=+1$<br>$\sigma_{2,4}=-1$<br>$\rho_{2,3}=+1$ | $\pi/2$<br>$\sigma_{1,10}=+1$<br>$\sigma_{2,2}=-1$<br>$\rho_{2,3}=-1$ | $\pi/2$<br>$\sigma_{1,3}=-1$<br>$\sigma_{2,4}=-1$<br>$\rho_{2,1}=+1$ | $\sigma_{1,1}=-1$<br>$\sigma_{2,4}=-1$<br>$\sigma_{3,4}=-1$<br>$\sigma_{3,6}=-1$ | $-\pi/2$<br>$\sigma_{1,7}=+1$<br>$\sigma_{2,4}=+1$<br>$\rho_{2,5}=-1$ | $\sigma_{1,8}=-1$<br>$\sigma_{2,2}=-1$<br>$\rho_{2,8}=+1$ | $\pi/2$<br>$\sigma_{1,7}=-1$<br>$\sigma_{2,6}=-1$<br>$\rho_{2,4}=-1$ | $\sigma_{1,1}=-1$<br>$\sigma_{1,5}=-1$ |

$$\hat{r}_0=\hat{r}_{0,a}=\hat{r}_{0,s}=2^{-4}+2^{-7}+2^{-10}+2^{-12}+2^{-15}-2^{-19}-2^{-21}$$
$$\hat{r}_1=\hat{r}_{1,a}=\hat{r}_{1,s}=-2^{-4}-2^{-5}+2^{-9}+2^{-12}+2^{-14}-2^{-17}+2^{-19}+2^{-20}$$
$$\hat{r}_2=\hat{r}_{2,a}=\hat{r}_{2,s}=-2^{-2}+2^{-5}-2^{-12}-2^{-14}-2^{-16}+2^{-19}$$
$$\hat{r}_3=\hat{r}_{3,a}=\hat{r}_{3,s}=-2^{-3}+2^{-8}-2^{-11}-2^{-12}+2^{-15}+2^{-18}$$

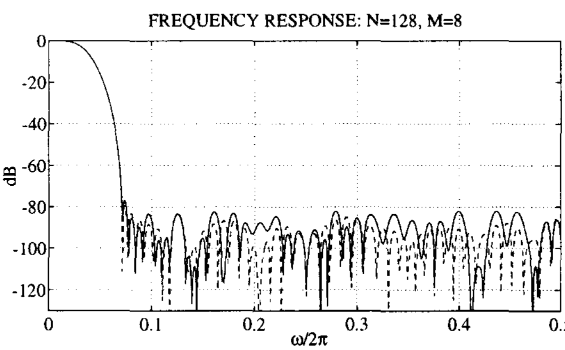


Figure 7: Frequency responses of original prototype [3] (dashed line) and efficient realization (solid line); $N=128$, $M=8$.

## 5. CONCLUSION

In this paper we have presented a simple procedure to find efficient realizations for prototype filters starting from a given PR prototype. Instead of a multiplication per coefficient we only have to perform a small number of shift and add operations. The resulting implementation, based on the lattice structure of the polyphase filters, is highly regular and allows parallel processing of all $M$ lattices. Although we do not necessarily find the best implementation since we do not optimize over the restricted parameter space for the $\mu$-rotations, we have shown that powerful results can be obtained by a straightforward algorithm at low computational cost. An efficient realization of the cosine modulation also based on $\mu$-rotations is described in [8].

**Acknowledgment**

The authors like to thank P. Rieder for helpful discussions.

## 6. REFERENCES

[1] N. J. Fliege. Computational Efficiency of Modified DFT-Polyphase Filter Banks. In *Proc. 27th Asilomar Conference on Signals, Systems and Computers*, 1993.

[2] P. P. Vaidyanathan. *Multirate Systems and Filter Banks.* Prentice Hall, Englewood Cliffs, 1993.

[3] T. Q. Nguyen. Digital Filter Banks Design - Quadratic-Constrained Formulation. *IEEE Trans. on Signal Processing*, SP-43:2103–2108, September 1995.

[4] J. E. Volder. The CORDIC Trigonometric Computing Technique. *IRE Trans. Electr. Comput.*, EC-8:330–334, 1959.

[5] P. Rieder, K. Gerganoff, J. Götze, and J. A. Nossek. Parametrization and Implementation of Orthogonal Wavelet Transform. In *Proc. IEEE ICASSP*, 1996.

[6] J. Götze and G. J. Hekstra. An Algorithm and Architecture Based on Orthonormal $\mu$-rotations forComputing the Symmetric EVD. *Integration- The VLSI Journal 475*, 1996.

[7] R. D. Koilpillai and P. P. Vaidyanathan. Cosine-Modulated FIR Filter Banks Satisfying Perfect Reconstruction. *IEEE Trans. on Signal Processing*, SP-40:770–783, April 1992.

[8] P. Rieder, J. Götze, and J. A. Nossek. Orthogonal Approximation of the Discret Cosine Transform. In *Proc. Eur. Conf. on Circuit Theory and Design*, 1995.