

Improving Concurrency for Cosine-modulated Filterbank Windowing

C.G. Hiremath and S. Jayasimha
Signion Systems Pvt. Ltd.
Hyderabad, INDIA

Abstract

A novel method exploits the time-reversal of mirror symmetric pairs of polyphase components of a linear-phase prototype filter to obtain more than 25% reduction in required MIPS to perform prototype windowing of either an analysis or synthesis cosine modulated filterbank on a suitable architecture. The architecture required to obtain this MIPS reduction is only a minor modification to many DSP architectures and is, in fact, available in the latest DSP designs. The proposed algorithm is particularly suited for large overlap factors, where the windowing computation dominates the required MIPS., one important example being the filterbank used in MPEG audio compression.

I Introduction

Of all filterbank structures, the cosine modulated filter banks is the most attractive with respect to both implementation cost and design ease, as the constituent filters are derived from the cosine modulation of a single prototype. One major application, the ISO/ MPEG subband coding of digital audio, utilizes a cosine modulated filter bank. The MPEG audio coding window corresponds to Type-II modulation (i.e., the window-length is odd, but made even with the introduction of an appropriate odd number of leading zeroes). The impulse response of the analysis and synthesis filters $h_k(n)$ and $f_k(n)$ are cosine modulated versions of the prototype filter $h(n)$. For a Type-II M -channel cosine modulated filter bank the analysis and synthesis filters are:

$$\begin{cases} h_k(n) = 2h(n) \cos\left((2k+1)\frac{\pi}{2M}\left(n - \frac{N}{2}\right) - \phi_k\right), \\ f_k(n) = 2h(n) \cos\left((2k+1)\frac{\pi}{2M}\left(n - \frac{N}{2}\right) + \phi_k\right), \end{cases} \begin{cases} 0 \leq n \leq N-1 \\ 0 \leq k \leq M-1 \end{cases} \quad (1a)$$

where, N is the length of the prototype filter $h(n)$, ϕ_k is the phases of the modulation index.

Another type of modulation function is Type IV modulation for which N in the modulation function of (1) is replaced by the order of the prototype (i.e., $N-1$). The impulse responses of the analysis and synthesis filters for Type IV modulation are

$$\begin{cases} h_k(n) = 2h(n) \cos\left((2k+1)\frac{\pi}{2M}\left(n - \frac{N-1}{2}\right) - \phi_k\right), \\ f_k(n) = 2h(n) \cos\left((2k+1)\frac{\pi}{2M}\left(n - \frac{N-1}{2}\right) + \phi_k\right), \end{cases} \begin{cases} 0 \leq n \leq N-1 \\ 0 \leq k \leq M-1 \end{cases} \quad (1b)$$

One choice for the phase of the modulation, ϕ_k , for adjacent channel alias cancellation is:

$$\phi_k = (2k+1)(2mM+1)\frac{\pi}{4} \quad (2)$$

$$\text{where } m = \left\lceil \frac{N}{2M} \right\rceil.$$

To implementation the filterbank efficiently using a DCT, the prototype filter is padded with zeros on either side from its mid point such that m is even. With this constraint on the length of the filter the phase, ϕ_k , simplifies to

$$\phi_k = (2k+1)\frac{\pi}{4} \quad (3)$$

Using (3) in (1a) the impulse responses of analysis and synthesis filters for Type II modulation is expressed as:

$$\begin{cases} h_k(n) = 2h(n) \cos\left((2k+1)\frac{\pi}{2M}\left(n - \frac{N}{2}\right) - (2k+1)\frac{\pi}{4}\right), \\ f_k(n) = 2h(n) \cos\left((2k+1)\frac{\pi}{2M}\left(n - \frac{N}{2}\right) + (2k+1)\frac{\pi}{4}\right), \end{cases} \quad (4)$$

$$\begin{cases} 0 \leq n \leq N-1 \\ 0 \leq k \leq M-1 \end{cases}$$

The impulse response for analysis filters can be separated into $2M$ polyphase components:

$$H_k = \sum_{i=0}^{2M-1} G_i(-z^{2M}) \cos\left((2k+1)\left(i - \frac{N}{2}\right)\frac{\pi}{2M} - (2k+1)\frac{\pi}{4}\right) z^{-i}$$

where $G_i(-z^{2M}) = \sum_{q=0}^{N-1} 2h(2qM+i)(-z)^{-2qM}$ (5)

$$\begin{cases} 0 \leq q \leq \frac{N}{2M} - 1 \\ 0 \leq k \leq M-1 \end{cases}$$

A fast algorithm for implementing this filterbank, using an IDCT-II like modulation matrix (the modulation matrix used is the same as the IDCT-II matrix for all coefficients other than the dc coefficient; for the dc coefficient, the scaling constant is 1 instead of the usual scaling factor), can be obtained by resequencing the polyphase components as follows:

$$G'_i = \begin{cases} z^{-\frac{M}{2}} G_{i+\frac{M}{2}} & \text{for } i = 0, 1, \dots, \frac{3M}{2} - 1 \\ -z^{-\frac{3M}{2}} G_{i-\frac{3M}{2}} & \text{for } i = \frac{3M}{2}, \dots, 2M-1 \end{cases} \quad (6)$$

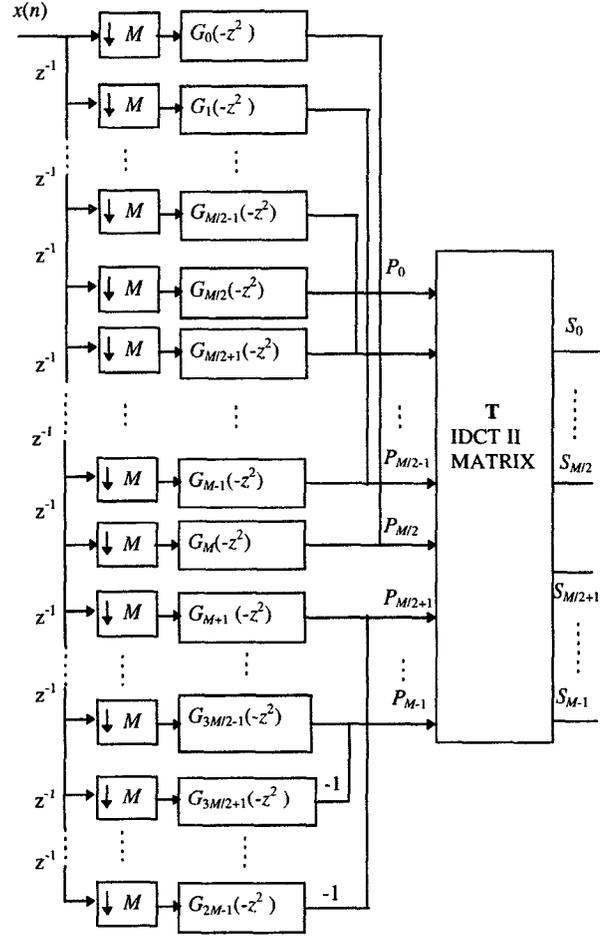


Figure 1. Cosine modulated analysis filter bank in terms of Type-II modulation.

In terms of this new sequence, the analysis filters are expressed as follows:

$$H_k(z) = G'_0(-z^{2M}) \cos\left((2k+1)\frac{\pi}{2M}\right) + \sum_{i=1}^{M-1} (z^{-i} G'_i(-z^{2M}) - z^{-(2M-i)} G'_{2M-i}(-z^{2M})) \cos\left((2k+1)\frac{\pi}{2M}\right) \quad (7)$$

$$0 \leq k \leq M-1$$

In matrix notation, the analysis filter bank vector $\mathbf{h}(z)$ becomes :

$$\mathbf{h}(z) = \mathbf{T}\mathbf{g}(z), \quad (8)$$

where,

$$\mathbf{h}(z) = \begin{bmatrix} H_0(z) \\ H_1(z) \\ \dots \\ H_{M-1}(z) \end{bmatrix},$$

$$\mathbf{g}(z) = \begin{bmatrix} G'_0(-z^{2M}) \\ z^{-1}G'_1(-z^{2M}) - z^{-(2M-1)}G'_{2M-1}(-z^{2M}) \\ \dots \\ z^{-M-1}G'_{M-1}(-z^{2M}) - z^{-(M+1)}G'_{M+1}(-z^{2M}) \end{bmatrix}$$

and \mathbf{T} is a $M \times M$ IDCT-II modulation matrix with elements:

$$t_{ki} = 2 \cos \left((2k+1)i \frac{\pi}{2M} \right)$$

A polyphase structure, structure, similar to that in [1], that implements the analysis filterbank is shown in Figure 1. For Type IV cosine modulated filter bank, the fast method uses IDCT IV like modulation matrix. The polyphase structure for Type IV cosine modulated filterbank is shown in Figure 2. The analysis filterbank vector is expressed as:

$$\mathbf{h}(z) = \mathbf{T}\mathbf{g}(z), \quad (9)$$

where,

$$\mathbf{h}(z) = \begin{bmatrix} H_0(z) \\ H_1(z) \\ \dots \\ H_{M-1}(z) \end{bmatrix},$$

$$\mathbf{g}(z) = \begin{bmatrix} G'_0(-z^{2M}) - z^{-(2M-1)}G'_{2M-1}(-z^{2M}) \\ z^{-1}G'_1(-z^{2M}) - z^{-(2M-2)}G'_{2M-2}(-z^{2M}) \\ \dots \\ z^{-(M-1)}G'_{M-1}(-z^{2M}) - z^{-M}G'_M(-z^{2M}) \end{bmatrix}$$

and \mathbf{T} is a $M \times M$ IDCT IV like modulation matrix with elements :

$$t_{ki} = 2 \cos \left((2k+1)(2i+1) \frac{\pi}{4M} \right)$$

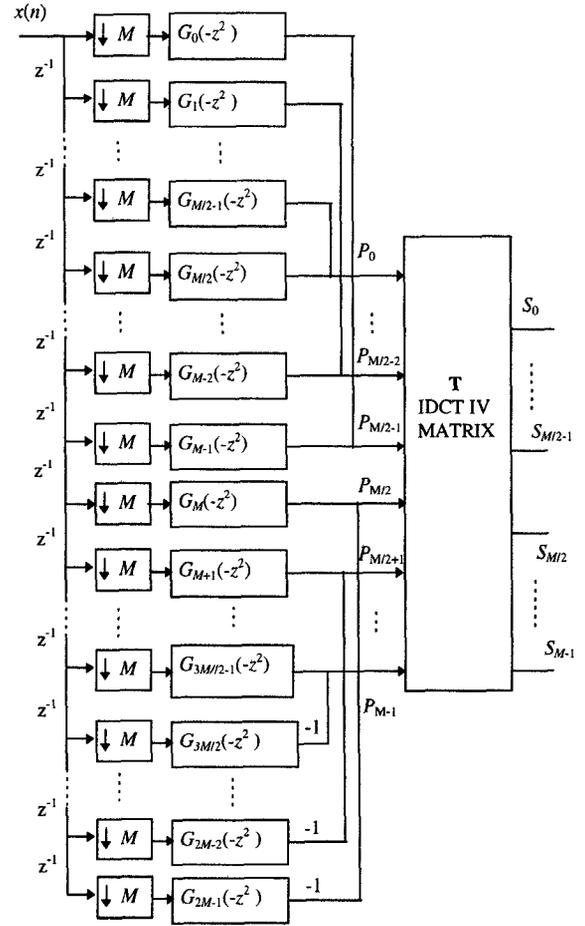


Figure 2. Type IV Cosine modulated analysis filter bank in terms of IDCT IV

II Improved Windowing

A new, improved windowing technique, for architectures of the type shown in Figure 3, is presented. This technique can be used to compute polyphase components P_1 to $P_{(M/2)-1}$ and from $P_{(M/2)+1}$ to P_{M-1} for Type II modulation and all the polyphase components P_0 to P_{M-1} for Type IV modulation. Note that the polyphase filter $G_{2M-i}(-z^{2M})$ is the mirror image of $G_i(-z^{2M})$ denoted by $\tilde{G}_i(-z^{2M})$. Note that $P_{(M/2)-i}$ and $z^M P_{(M/2)+i}$ are related by the matrix equation:

$$\begin{aligned} \begin{bmatrix} P_{\frac{M}{2}-i} \\ z^{\frac{M}{2}} P_{\frac{M}{2}+i} \end{bmatrix} &= \begin{bmatrix} G_i(-z^{2M}) & G_{M-i}(-z^{2M}) \\ G_{M+i}(-z^{2M}) & -G_{2M-i}(-z^{2M}) \end{bmatrix} \begin{bmatrix} X_i \\ X_{M-i} \end{bmatrix} \\ &= \begin{bmatrix} G_i(-z^{2M}) & G_{M-i}(-z^{2M}) \\ \tilde{G}_{M-i}(-z^{2M}) & -\tilde{G}_i(-z^{2M}) \end{bmatrix} \begin{bmatrix} X_i \\ X_{M-i} \end{bmatrix}, \quad (10) \\ &1 \leq i \leq \frac{M}{2} - 1 \end{aligned}$$

where, $X_i = [x_i \ z^{-2M}x_i \ z^{-4M}x_i \ z^{-6M}x_i \ \dots \ z^{-N-2M}x_i]$ and $X_{M-i} = [x_{M-i} \ z^{-2M}x_{M-i} \ z^{-4M}x_{M-i} \ z^{-6M}x_{M-i} \ \dots \ z^{-N-2M}x_{M-i}]$

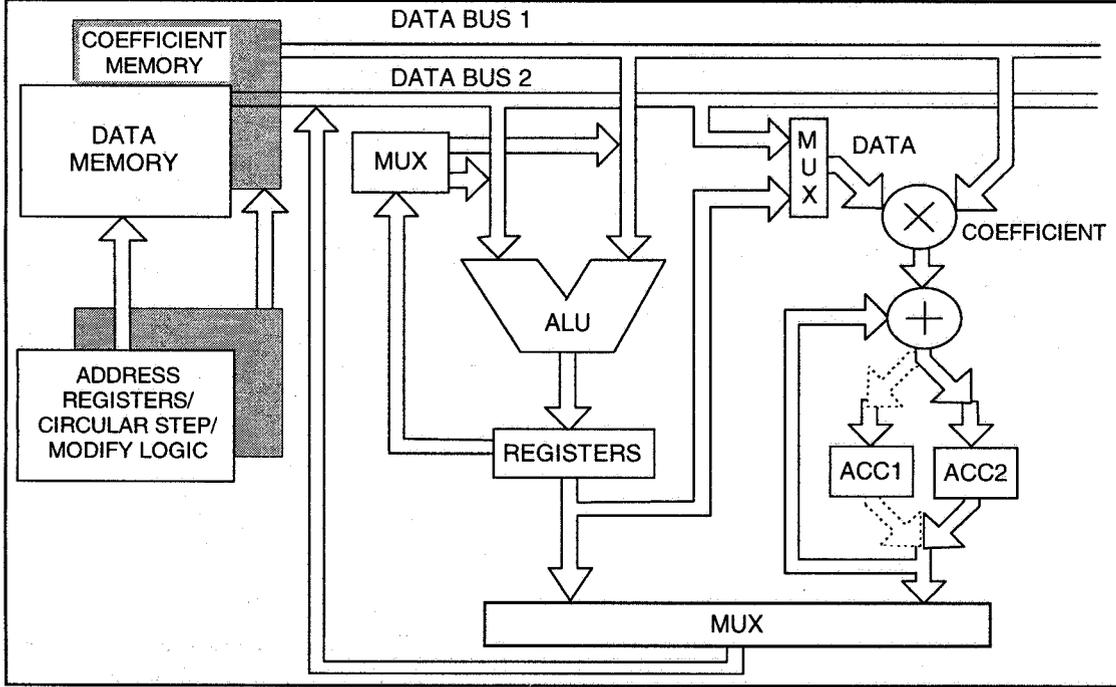


Figure 3. Architecture that exploits (14) to reduce required MIPS.

Direct computation of the polyphase components $P_{(M/2)-i}$ and $z^M P_{(M/2)+i}$ using the SIMD architecture requires $2N/M$ multiplications and $(2N/M)-2$ additions. This SIMD architecture has all the features shown in Figure 3 except that a single accumulator replaces the dual accumulator and the ALU need not execute in parallel with the multiply accumulator (MAC).

To reduce the instruction rate, (10) can be expressed as a sum of butterfly operations:

$$\begin{aligned} \begin{bmatrix} P_{\frac{M}{2}-i} \\ z^{\frac{M}{2}} P_{\frac{M}{2}+i} \end{bmatrix} &= \begin{bmatrix} g_i & z^{-(N-M)} g_{M-i} \\ -z^{-(N-M)} g_{M-i} & g_i \end{bmatrix} \cdot \begin{bmatrix} x_i \\ z^{-(N-M)} x_{M-i} \end{bmatrix} + \\ &\dots + \begin{bmatrix} z^{-(N-M)} g_i & g_{M-i} \\ -g_{M-i} & z^{-(N-M)} g_i \end{bmatrix} \begin{bmatrix} z^{-2M} x_i \\ z^{-(N-4M)} x_{32-i} \end{bmatrix} \\ &= \begin{bmatrix} P'_{\frac{M}{2}-i,0} \\ z^{\frac{M}{2}} P'_{\frac{M}{2}+i,0} \end{bmatrix} + \dots + \begin{bmatrix} P'_{\frac{M}{2}-i,\frac{M-1}{4}} \\ z^{\frac{M}{2}} P'_{\frac{M}{2}+i,\frac{M-1}{4}} \end{bmatrix} \end{aligned} \quad (11)$$

One simple optimization technique is to compute each butterfly using 3 multiplies and 3 adds [2] as shown in Figure. 3 instead of the 4 multiplies and 2 adds implied by (11).

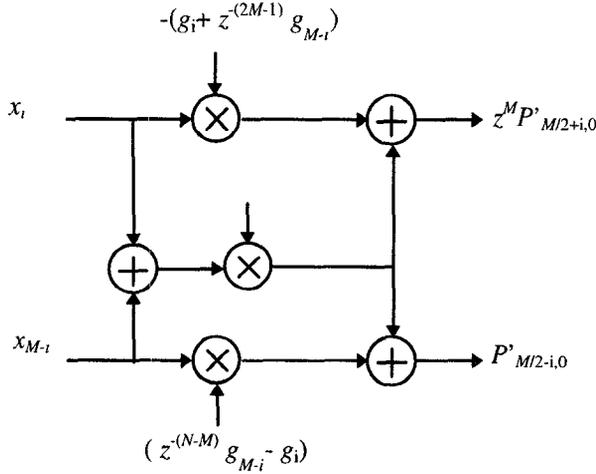


Figure.4 Butterfly operation implemented with three multiplies and three adds.

For MPEG audio windowing ($M=32$, $N=511$), this method requires 24 multiplies and 24 additions for computing all the butterflies for each pair of polyphase components. Another 14 additions are required to accumulate the outputs of the butterflies. In architectures where the computation of the butterflies and their summation is a parallel operation, this method yields about twenty five percent reduction in computational complexity. However, an alternative regular structure, that uses the simpler architecture of Figure 3, provides a slightly superior complexity reduction.

First, express (10) as the sum of a single (rather than dual) polyphase component and an exact mirror image structure [3] as follows:

$$\begin{aligned} \begin{bmatrix} P_{M-i} \\ z^{\frac{M}{2}} P_{M+i} \end{bmatrix} &= \begin{bmatrix} 0 & 0 \\ 0 & -2\tilde{G}_i(-z^{2M}) \end{bmatrix} \begin{bmatrix} X_i \\ X_{M-i} \end{bmatrix} \\ &+ \begin{bmatrix} G_i(-z^{2M}) & G_{M-i}(-z^{2M}) \\ \tilde{G}_{M-i}(-z^{2M}) & \tilde{G}_i(-z^{2M}) \end{bmatrix} \begin{bmatrix} X_i \\ X_{M-i} \end{bmatrix} \\ &= \begin{bmatrix} P''_{M-i} \\ z^M P''_{M+i} \end{bmatrix} + \begin{bmatrix} P'_{M-i} \\ z^M P'_{M+i} \end{bmatrix} \end{aligned} \quad (12)$$

The second part of the above equation is detailed in (13), while the complexity associated with (13) can be reduced by a factor of 2 by the equivalent form [3] given by (14).

$$\begin{aligned} \begin{bmatrix} P'_{M-i} \\ z^M P'_{M+i} \end{bmatrix} &= \begin{bmatrix} g_i \dots z^{-(N-M)} g_i & g_{M-i} & \dots & z^{-(N-M)} g_{M-i} \\ z^{-(N-M)} g_{M-i} \dots g_{M-i} & g_{M-i} & z^{-(N-M)} g_i & \dots g_i \end{bmatrix} \\ &\begin{bmatrix} x_i \\ \dots \\ z^{-(N-M)} x_i \\ x_{M-i} \\ \dots \\ z^{-(N-M)} x_{M-i} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} c_0 \dots c_{\frac{M-1}{4}} & 0 \dots 0 \\ 0 \dots 0 & d_{\frac{M-1}{4}} \dots d_0 \end{bmatrix} \begin{bmatrix} x_i + z^{-(N-M)} x_{M-i} \\ \dots \\ z^{-(N-M)} x_i + x_{M-i} \\ z^{-(N-M)} x_i - x_{M-i} \\ \dots \\ x_i - z^{-(N-M)} x_{M-i} \end{bmatrix} \end{aligned} \quad (13)$$

$$(14)$$

$$\begin{aligned} \text{where, } c_i &= (z^{-2M} g_i + z^{-(N-2M)+2M} g_{M-i})/2, \\ d_i &= (z^{-2M} g_i - z^{-(N-2M)+2M} g_{M-i})/2. \end{aligned}$$

For MPEG audio coding, the computation of the first part of (12) requires 8 multiplications and seven additions per polyphase component pair. The computation of the second part of (12) is done via (14) utilizing 16 multiplications 30 additions and one add/subtract operation per polyphase component pair. Finally, one more addition is needed to sum $P'_{M/2+i}$ and $P''_{M/2+i}$. For the architecture of Fig. 3, the number of instructions executed per polyphase component pair is 26 as compared to the 32 instructions consumed by the direct realization (yielding a savings in required MIPs by 25%). For any window length, N , and any number of subbands, M , is (asymptotically) $1.5 \cdot N/M$ instructions per polyphase component pair are consumed. A small additional complexity reduction is obtained by utilizing the fact that $G_0(-z^2)$ and $G_M(-z^2)$ are symmetric. Since these filter outputs are added to compute the polyphase component, $P_{M/2}$, they can be combined into a single linear phase filter as shown by the lower instruction-rate program of Figure A.2 in the appendix (for the architecture of Figure 3).

A similar reduction can also be obtained for the synthesis filterbank of Figure 5 by noting that $x_i(n)$ and $x_{M-i}(n)$ are related to the polyphase components by the matrix equation:

$$\begin{aligned}
\begin{bmatrix} x_{si} \\ x_{sM-i} \end{bmatrix} &= \begin{bmatrix} \tilde{G}_i(-z^{2M}) & -\tilde{G}_{i+M}(-z^{2M}) \\ -\tilde{G}_{M-i}(-z^{2M}) & -\tilde{G}_{2M-i}(-z^{2M}) \end{bmatrix} \begin{bmatrix} P_{\frac{M}{2}+i} \\ z^{-1}P_{\frac{M}{2}-i} \end{bmatrix} = \begin{bmatrix} \tilde{G}_i(-z^{2M}) & -G_{M-i}(-z^{2M}) \\ -\tilde{G}_{M-i}(-z^{2M}) & -G_i(-z^{2M}) \end{bmatrix} \begin{bmatrix} P_{\frac{M}{2}+i} \\ z^{-1}P_{\frac{M}{2}-i} \end{bmatrix} \\
&= \begin{bmatrix} 2\tilde{G}_i(-z^{2M}) & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P_{\frac{M}{2}+i} \\ z^{-1}P_{\frac{M}{2}-i} \end{bmatrix} + \begin{bmatrix} -\tilde{G}_i(-z^{2M}) & -G_{M-i}(-z^{2M}) \\ -\tilde{G}_{M-i}(-z^{2M}) & -G_i(-z^{2M}) \end{bmatrix} \begin{bmatrix} P_{\frac{M}{2}+i} \\ z^{-1}P_{\frac{M}{2}-i} \end{bmatrix}, \quad 1 \leq i \leq \frac{M}{2} - 1
\end{aligned} \tag{15}$$

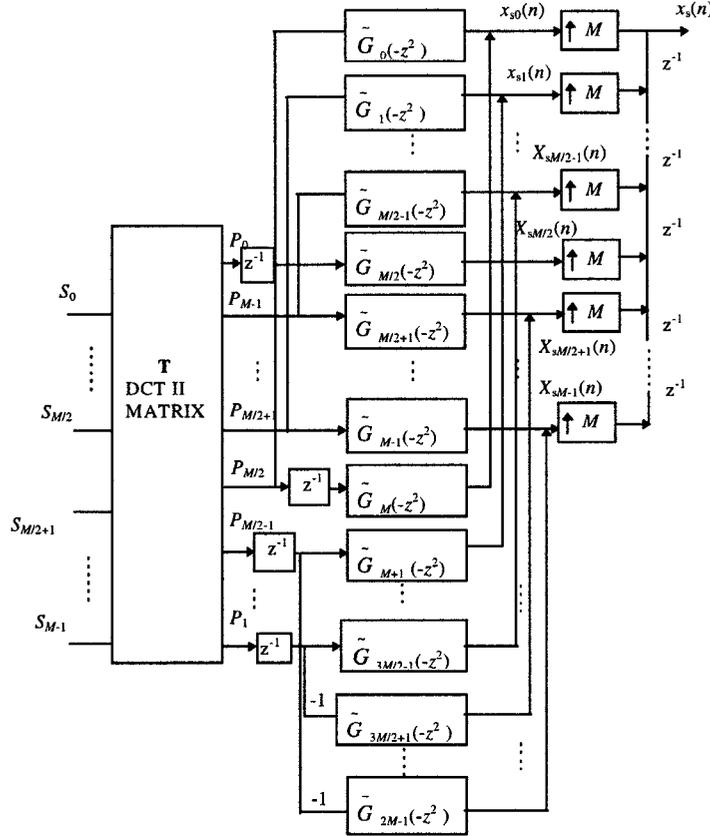


Figure 5. Cosine modulated synthesis filterbank in terms of Type-II modulation.

III Conclusion

A regular algorithmic structure for MPEG audio analysis and synthesis windowing that, for an architecture already implemented on some DSPs, achieves a 25% reduction in required MIPS is described. The required architecture has only a very small incremental gate count over a conventional SIMD DSP architecture, requiring only one additional MAC accumulator register and two additional multiplexers. The impact of this algorithmic improvement is significant in the implementation of

multichannel extensions of MPEG and other audio coding standards.

References

- [1] P. P. Vaidyanathan, "Multirate Systems and Filter Banks," pp. 388-391, Prentice Hall Signal Processing Series, 1993.
- [2] H. S. Malvar, "Signal processing with Lapped Transforms," pp. 60, Artech House, 1992.
- [3] Z.-J. Mou, "Symmetry Exploitation in Digital Interpolators/Decimators", pp. 2611-2615, *IEEE trans. on SP*, Vol. 44, No. 10, Oct 1996.