EFFICIENT DESIGN OF A CLASS OF MULTIPLIER-LESS PERFECT RECONSTRUCTION TWO-CHANNEL FILTER BANKS AND WAVELETS WITH PRESCRIBED OUTPUT ACCURACY

Carson, K.S. Pun, S.C. Chan and K.L. Ho

Department of Electrical and Electronic Engineering,
The University of Hong Kong

Email: kspun@eee.hku.hk, scchan@eee.hku.hk, klho@eee.hku.hk

ABSTRACT

This paper proposes a novel algorithm for the design and hardware reduction of a class of multiplier-less two-channel PR filter banks (FBs) using sum-of-powers-of-two (SOPOT) coefficient. It minimizes a more realistic hardware cost, such as adder cells, subject to a prescribe output accuracy taking into account of the rounding and overflow effects, instead of using just the SOPOT terms as in conventional method. Furthermore, by implementing the filters in the FBs using multiplier-block (MB), significant overall saving in hardware resources can be achieved. An effective random search algorithm is also proposed to solve the design problem, which is also applicable to PR IIR FBs with highly nonlinear objective functions.

I. INTRODUCTION

Perfect reconstruction (PR) multirate filter banks (FB) have important applications in signal analysis, signal coding and the design of wavelet bases. A number of techniques for designing linear-phase and low-delay PR two-channel filter banks are now available [1][2][3]. Recently, there is an increasing interest in designing PR filter banks with very low implementation complexity. One of the applications is to provide efficient hardware implementation of the 9/7 wavelet filter for the JPEG2000 standard. FBs using sum of powers-of-two (SOPOT) coefficients are particularly attractive for VLSI or hardware implementation because multiplication of SOPOT coefficients can be implemented efficiently using hard-wired shifters and adders only (i.e. multiplier-less). The design of such SOPOT PR FBs using the 2-channel lossless lattice structure and genetic algorithm was studied in [6]. Another family of multiplier-less PR two-channel FIR/IIR FB and wavelets, using SOPOT coefficients and the structure in [1], was studied recently by the authors in [4][7]. They are attractive because of their low hardware and design complexities. Furthermore, the PR condition is structurally imposed and is robust to coefficient quantization.

It is well known that there are two sources of error in implementing a digital filter: coefficient round-off error and signal round-off error [10]. Coefficient round-off error happens when the real-valued coefficients of the filter, obtained say by the Park-McClellan algorithm, are rounded to their fixed-point representations to simply the hardware implementation. The frequency response of the filter is therefore changed, and might not satisfy the specification any more. On the other hand, signal round-off error occurs when overflow occurs due to insufficient internal wordlength and improper scaling; and when rounding is performed for long intermediate data after multiplications with the filter coefficients. Signal round-off error is usually more difficult to handle in hardware implementation because complicated hardware for detecting overflows, etc., would significantly slow down the throughput of the system. The SOPOT FBs mentioned above are free from coefficient round-off noise because the FBs are optimized using the SOPOT coefficients as variables. Unfortunately, most of these methods only focused on minimizing the number of SOPOT terms to meet a given frequency specification, and pay little attention to signal round-off error. In order to satisfy a given output accuracy, one usually employs a fixed and long wordlength for all intermediate data, which means increased hardware complexity. Therefore, the design problem should be to minimize the hardware complexity of the system while satisfying the given frequency specification and the output accuracy. The hardware complexity could be the number of adder cells and registers used in the FBs, which is related to the exact wordlength used for each intermediate data. The output accuracy of a digital filter is usually specified statistically by its output noise power due to the rounding operations performed, using a given noise model. For fine quantization, round-off noise is usually modeled as white and is uncorrelated with the signal and other noise sources. To satisfy a given output accuracy (say 16-bit), one has to determine the appropriate scaling and wordlength of each intermediate data to avoid signal overflow and to achieve a noise power less than the given specification (say -96dB for 16-bit accuracy).

The purpose of this paper is to provide a solution to the above problem, with particular emphasis on the SOPOT FBs that we have proposed in [4][7]. This class of PR FBs is chosen because the required stopband attenuation and system delay can easily be achieved using simple design formula for order estimation and the efficient Park-McClellan design algorithm. Using the real-valued coefficients so obtained as initial guess, the SOPOT coefficients and the internal wordlength of all intermediate data are jointly optimized using a novel random search algorithm to minimize some measure of the hardware complexity, while satisfying the given specification. In this work, both the number of adders and their adder cells are minimized because they constituted over 70% of the total hardware cost as compared with other components such as latches. The random search algorithm is similar to the mutation of genetic algorithm (GA) and the random walk in stimulated annealing. The main difference here is that we have limited its search space to a small neighborhood of the real-valued solution obtained in [4][7] using the Park-McClellan algorithm. This greatly shortens the search time to a few minutes. Moreover, for IIR FBs, excellent SOPOT solutions can be obtained in a reasonably time, which cannot be achieved by GA even with design time several orders of magnitude longer. The latter is mainly due to high sensitivities of the poles. The number of adders required to implement the SOPOT multiplications is further reduced by using the technique of "multiplier-block" (MB) [9]. By using MB, redundancy in the SOPOT coefficients is removed. Design examples demonstrated that our design method is very efficient and capable of reducing dramatically the hardware complexity of the FBs, while meeting the given specifications. More difficult 2-channel SOPOT IIR PR FBs can also be designed using the proposed method. Our paper is organized as follows: in section II, the SOPOT FBs considered and the MB technique will be described. The round-off-noise and overflow problems will be addressed in Section III. Section IV is devoted to the 'Random search' design algorithm. This is followed by several design examples in Section V. Finally conclusions are drawn in section VI.

II. 2-CHANNEL PR SOPOT FB

Fig. 1 shows the structure of the PR FB proposed in [5]. The functions $\alpha(z)$ and $\beta(z)$ can be linear-phase FIR, nonlinear-phase FIR, or IIR functions, without affecting the PR conditions. It can be shown that the lowpass and highpass analysis filters are given by $H_0(z)=(z^{-2N}+z^{-1}\beta(z^2))/2$ and $H_1(z)=-\alpha(z^2)H_0(z)+z^{-2M-1}$, respectively. It is also possible to realize wavelet bases from this FBs by imposing certain regularity condition on $H_0(z)$ and $H_1(z)$. Details regarding their design can be found in [4]. In the multiplier-less FB [4][7], each coefficient in $\alpha(z)$ and $\beta(z)$ is represented as the following sum of powers-of-two coefficients (SOPOT) or canonical signed digits (CSD), $b=\sum_{k=0}^{L-1} a_k \cdot 2^{-b_k}$, where a_k is either 1 or −1, and $b_k \in \{-l_L,\ldots 1,0,\ldots l_U\}$. The larger the numbers l_L, l_U, and L, the closer the SOPOT approximation will be to the original real number. In practice, the number of non-zero terms is usually kept to a small number while satisfying a given specification so that the multiplication can be implemented as a limited number of shift and add (subtract) operations, giving rise to multiplier-less realization. Multiplier-less filter banks and wavelet bases with linear-phase and low system delay can be obtained from this structure by searching for the SOPOT coefficients using the genetic algorithm [6][7]. As mentioned earlier, the number of adders needed to implement $\alpha(z)$ and $\beta(z)$ can further be reduced

0-7803-7011-2/01/$10.00 ©2001 IEEE

by rewriting them in transposed form. It can be seen that instead of multiplying the delayed input samples with the filter coefficients as in the direct form, the input sample is now multiplied with all the coefficients. This can be efficiently implemented using a multiplier block (MB) [9]. Let's consider a simple example with two filter coefficients: 3 and 21. The SOPOT representations of these two numbers are: $3=2^1+1$ and $21=2^4+2^2+1$. This requires 3 adders and 3 shifts. If implemented in a MB, the multiplication of the input with the coefficient 3 will also be generated by decomposing 3 as 2^1+1, requiring one addition. The multiplication with 21, however, can be simplified by re-using the intermediate result generated by the first filter coefficient '3' as $21=3\cdot 7=3\cdot(2^4-1)$. Actually, the intermediate result, after multiplication by 3, is multiplied by 7, which requires one less adder than generating 21 directly. In principle, it is possible to remove all the redundancy found in the constant multipliers leading to a realization with the *minimum number of adders*. This can drastically reduce the number of adders required for realizing such FBs when there is a large number of filter coefficients to be implemented in the transposed form FIR structure (around 50% in our example).

III. Round-Off Noise and Overflow Analyses

1. Analysis of Round-off Noise

As mentioned earlier, round-off noise occurs when rounding is performed during arithmetic computation. In fixed-point arithmetic, round-off operation is usually performed after multiplication to limit the wordlength of the intermediate data in order to save hardware resources. Round-off error is thus generated. Due to the difficulty in analyzing exactly the rounding error, they are usually treated as white random process, uncorrelated with the signal and other noise sources. For rounding operation, quantization noise will have zero mean and a variance $\sigma^2=\Delta^2/12$, where Δ is the quantization step-size, which is determined by the number of fractional bits that is retained after multiplication.

Consider the transposed form FIR filter in figure 2. The blocks **D** and **Q{.}** represent respectively a register and the round-off operator. Any signal in this filter, for example the input signal $x[n]$, has a fixed-point representation of the form $<n|m>$, which means that the total wordlength is $n+m$ bits where n represents the integer bits (including the sign bit) and m the fractional bits. For notation convenience, any signal will be represented as $x[n]:<n|m>$, meaning that it has n integer bits and m fractional bits. Now, consider the input sample $x[n]:<1|7>$, which is a 8-bit number gated into the digital filter at every clock cycle. It will be multiplied by $h[0]:<1|9>$ and $h[1]:<1|7>$. If no rounding is performed, the fixed-point formats of the products $x[n]h[0]$ and $x[n]h[1]$ will be $<1|16>$ and $<1|14>$, respectively. Suppose that the products are rounded by the operator **Q{.}** to the format: $<1|14>$. Since the wordlength of $x[n]h[1]$ before and after rounding is equal, so there is no round-off noise ($e\_1[n]=0$). While for the signal $x[n]h[0]$, the wordlength is shortened from $<1|16>$ to $<1|14>$, hence, a round-off noise, $e\_0[n]\neq 0$, with a power of $P=(2^{-13})^2/12$ is generated. In general, if R, the number of bits in the fractional part of the fixed-point representation, is rounded to B $(B<R)$, then the round-off noise power is given by: $P_e=2^{-2(B-1)}/12$ [10]. If there are M such rounding noise sources in the transposed form, the total noise power at the output is given simply by their sum: $P_{total}=\sum_{k=1}^{M}P_{e_k}=\sum_{k=1}^{M}2^{-2(B_k-1)}/12$. For a general digital filter, the k*th* noise source might pass through a transfer function with z-transform $H_k(z)=\sum_{n=0}^{L_k}h(n)z^{-n}$, then the total output noise power is $P_{total}=\sum_{k=1}^{M}P_{e_k}|H(0)|^2$, assuming that they are uncorrelated. The output accuracy, in terms of the number of fractional bits, is therefore given by $(1/6)\cdot 10\cdot\log_{10}(P_{total})$. In general, to have 16-bit output accuracy, the output noise-power must be below –96 dB level. From these results, we can see that, the larger the number of noise sources, the lower will be the accuracy of the computation. The noise power can however be reduced by increasing the wordlength for the fractional bits, at the expense of increased hardware complexity.

2. Preventing Overflow

Another important source of error is signal overflow [10], which occurs when the allocated wordlength in the integer part is insufficient to represent correctly the fixed-point representation of the output after addition (such as the adders in Fig. 2). In order to avoid overflow, we must allocate more bits to the integer part of the register (say **D** in Fig.2). We are given the option to retain or decrease the number of bits in the fractional part, depending on the required accuracy. To determine whether overflow will occur for a given adder, we can compute certain measures of the transfer function from the input to this particular adder. Here, we prefer a more conservative measure using the absolute sum of the impulse response, i.e. L1 scaling. For example, let x_{max} be the maximum input to a FIR transposed form digital filter $H(z)=\sum_{k=0}^{L}h(k)z^{-k}$ as shown Fig. 2. Then the maximum (or worse case) value at the output of the l^{th} adders of the FIR filter is $d_l=\left(\sum_{k=1}^{l}|h(k)|\right)x_{max}, l=0,\ldots L$. From these values, it is possible to determine the required integer wordlength at each position to avoid any overflow. The number of fractional bits will be optimized to satisfy the given output accuracy. It should be noted that there are other scaling method such as L2 scaling which can also be used. However, there is still a small probability that overflow will occur. In digital signal processor, special hardware is usually used to detect the present of overflow and the result will be clipped to the maximum/minimum values of the representation (saturation arithmetic).

IV. The Design Algorithm

Our design method consists of two parts. First, the parameters of the filters $\alpha(z)$ and $\beta(z)$ such as their coefficients and their order (parameters N and M) are determined from the frequency specification (system delay, stopband attenuation, cutoff frequencies) using the method in [4]. Then, the SOPOT coefficients are determined using a random search algorithm to generate the MB (see 1 below). The hardware complexity of the FBs are then minimized while maintaining the output accuracy using the noise models mentioned earlier (see 2 below).

1. *Search for the SOPOT filter coefficients.*

The optimization procedure consists of two stages. First, a random search algorithm, to be discussed in the sequel, is used to search for the SOPOT coefficients of $\alpha(z)$ and $\beta(z)$ such that a given performance measure is minimized. Then, the minimum number of adders needed in the multiplier block is determined. The generation of the multiplier-block from the SOPOT coefficients follows the algorithms proposed in [9]. Let $\mathbf{x}_I$ be the vector containing the real-valued coefficients of $\alpha(z)$ and $\beta(z)$ obtained by the method in [4]. The principle of the random search algorithm is to generate random candidate SOPOT coefficients in the neighborhood of $\mathbf{x}_I$ so as to search for the optimal discrete solution. More precisely, a new coefficient vector $\mathbf{x}_{NEW}$ is generated by adding to it a random vector to the original coefficient vector $\mathbf{x}_I$ to form $\mathbf{x}_{NEW}=[\mathbf{x}_I+\alpha\cdot\mathbf{x}_R]_{SOPOT}$, where α is a scale factor which controls the size of the neighborhood to be searched, $\mathbf{x}_R$ is a vector with its elements being random numbers in the range $[-1,1]$, and $[\cdot]_{SOPOT}$ is the rounding operator which converts its argument to the nearest SOPOT coefficients with maximum number of terms in each coefficient being L and dynamic range l_U and l_L. The following objective function, which is the minimax error between the desired frequency response $H_d(e^{j\omega})$ and the

frequency response $H(e^{j\omega}, \hat{\mathbf{x}})$ calculated using the candidate $\hat{\mathbf{x}}$ in the frequency band of interest $\omega \in S$, is minimized:

$$score = \max_{\omega \in S} \left\{ \left| H(e^{j\omega}, \hat{\mathbf{x}}) - H_d(e^{j\omega}) \right| \right\}. \quad (1)$$

The process is repeated with different vector $\hat{\mathbf{x}}$ so that the SOPOT space in the neighborhood of $\hat{\mathbf{x}}$ is sampled randomly. Since the sampled solutions are close to the real-valued optimal solution, their frequency responses will also be close to the ideal one, but with different hardware complexity. The set that yields the minimum score with a given number of terms is recorded. As this is a random search algorithm, the longer the searching time, the higher the chance of finding the optimal solution.

2. Minimization of the filter banks hardware structures with prescribed output accuracy

After the MB is generated, the maximum wordlength of all the products, x[n]h[i], i=0,…,L, in Fig. 2, is calculated. If we do not perform any rounding using the operator $\mathbf{Q}\{.\}$, and sufficient wordlength is allocated to all adders, then there is no rounding error. Of course, this will require excessive hardware cost, especially when the output accuracy is low. Our goal is to determine the format of the rounded signals, $\mathbf{Q}\{x[n]h[i]\}$, i=0,…,L, to satisfy the output accuracy. Suppose that the formats are stored in a vector $\boldsymbol{\delta}$. Given the rounded output format of the MB, $\boldsymbol{\delta}$, one can determine, using the method described in Section III.2, the formats of the registers, **D**'s, and the structure of the adders, in order to avoid any overflow. The fractional part for those scaled output, to prevent overflow, can either retain its wordlengh or reduce it by one as mentioned in Section III.2. This option is stored in a vector $\boldsymbol{\delta}_f$, to be optimized together with $\boldsymbol{\delta}$. The noise power at the filter output of the filter is readily computed accordingly to the analysis described in Section III.1. Note the output noise power from $\alpha(z)$ and $\beta(z)$ will be evaluated and their contributions at the lowpass (and highpass) analysis filters will be properly summed, using their respective power transfer functions mentioned in Section III.1. Our design algorithm seeks to lower the wordlength of each intermediate data and hence the complexity format as specified in $\boldsymbol{\delta}$ and $\boldsymbol{\delta}_f$ to minimize the hardware cost. Using $\boldsymbol{\delta}$ and $\boldsymbol{\delta}_f$, the hardware cost, C, given by the adder cells in the MB and the subsequent adders in Fig. 2 can be evaluated. In summary, the design problem is

$$\min_{(\boldsymbol{\delta}, \boldsymbol{\delta}_f)} C(\boldsymbol{\delta}, \boldsymbol{\delta}_f) \text{ subject to } P_{total} \le P_{spec}, \quad (2)$$

where P_{total} is the output noise power at the lowpass and highpass filters and P_{spec} is the specified output accuracy. Using a random search algorithm similar to that mentioned in Section IV.1, the vector $(\boldsymbol{\delta}, \boldsymbol{\delta}_f)$ is searched in the neighborhood of their full precision values $(\boldsymbol{\delta}, \boldsymbol{\delta}_f)_\infty$ (that is no rounding) for feasible solutions that satisfying the given output accuracy. The one with the minimum hardware cost $C(\boldsymbol{\delta}, \boldsymbol{\delta}_f)$ is declared as the solution of this problem. There are several advantages of this algorithm. First of all, with the computational power of nowadays personal computer (PC) the time for obtaining high quality solutions is manageable, especially when an initial real-valued solution is available by some means. In fact, for the problem considered here, the overall design time is less than 10 minutes using a Pentium-400 PC with Matlab 5.3, including both the design of SOPOT coefficients, generation of the MB and the internal wordlength allocation. Secondly, it is applicable to problems with general objective functions probably with very complicated inequality constraints, as illustrated in this work. It is also possible to combine the search with the MB generation processes together for better performance but the computational time will be greatly increased. We now present a few design examples.

V. Design Examples

5.1. Two-channel PR FBs with $\beta(z)$ and $\alpha(z)$ FIR filters

To demonstrate the effectiveness of our algorithm for solving the complicated design problem, a two-channel FB with the following frequency specification is designed: passband and stopband cutoff frequencies ω_p= 0.4π, and ω_s=0.6π, respectively; stopband attenuation is 39 dB, system delay = 23. From the design procedure in [4], the parameters N and M are determined to be 3 and 8, respectively. The wordlength of the input is 8-bit and is normalized to be less than 1, i.e. in <1|7> format. The required output accuracy is at least 16-bit for fractional part without overflow. The frequency response of the final SOPOT FB is shown in figure 3, and the details of its optimized structure are summarized in table 1. The reduction of the number of adders obtained by using MBs to implement $\beta(z)$ and $\alpha(z)$ is around 50%. It can also be observed that the number of adder cells is significantly reduced by 27% (compared with a fixed wordlength of 24 bits using MBs to satisfying the same output accuracy) using the proposed random search algorithm to minimize the necessary internal wordlength, while satisfying the prescribed output accuracy of 16-bit. The overall design takes about 10 minutes on a typical Pentium-533 computer.

5.2. Two-channel PR FB with $\beta(z)$ IIR and $\alpha(z)$ FIR

To demonstrate the effectiveness of our random search algorithm in designing SOPOT PR IIR FB, $\beta(z)$ is chosen as an IIR filter while $\alpha(z)$ as an FIR filter. In order to guarantee the stability of the IIR filter, the denominator of $\beta(z)$ is factorized as a lattice structure and the magnitude of the lattice coefficients are forced to be less than 1. They are then used as optimization variable in the random search algorithm. The design specifications are: passband cutoff frequency ω_p =0.4π, stopband cutoff frequency ω_s =0.6π. N and M are determined to be 4 and *11*, respectively. The real-valued filter coefficients are obtained by the method in [11]. The SOPOT coefficients of the FBs obtained by the proposed algorithm are shown in table 2, and the frequency response is shown in figure 4. The frequency characteristic is very good despite the high nonlinearity of the objective function for the IIR FBs. From our experience, similar results cannot be achieved by GA even with design time several orders of magnitude longer. The latter is mainly due to high sensitivities of the poles. Since only the SOPOT coefficient optimization is performed, the computation time is much shorter, only 6 minutes in this case. The hardware structure is omitted here due to page length limitation.

VI. Conclusion

A novel algorithm for the design and hardware reduction of a class of multiplier-less two-channel PR FBs using SOPOT is presented. It minimizes a more realistic hardware cost, such as adder cells, subject to a prescribe output accuracy taking into account rounding and overflow effects. Further, by implementing the filters in the FBs using multiplier-block (MB), significant overall saving in hardware resources can be achieved. An effective random search algorithm is also proposed to solve the design problem, which is also applicable to PR IIR FB with highly nonlinear objective functions.

References

[1] W. Liu, S. C. Chan, and K. L. Ho, "Low-delay perfect reconstruction two-channel FIR/IIR filter banks and wavelet bases with SOPOT coefficients," in *Proc. IEEE ICASSP'2000, Istanbul, Turkey*, May 2000.

[2] G. Schuller and M. J. T. Smith, "A new algorithm for efficient low delay filter bank design," in *Proc. IEEE ICASSP'95*, vol. 2, pp. 1472 –1475, May. 1995.

[3] R. Gandhi and S. K. Mitra, "Design of two-channel low delay perfect reconstruction filter banks," in *Proc. 32nd International Asilomar Conference on Signals, Systems & Computers*, vol. 2, pp. 1655 –1659, Nov. 1998.

[4] J. S. Mao, S. C. Chan, W. Liu, and K. L. Ho, "Design and multiplier-less implementation of a class of two-channel PR FIR filterbanks and wavelets with low system delay," *IEEE Trans. SP.*, vol. 48, pp. 3379 – 3394, Dec. 2000.

[5] S. M. Phoong, C. W. Kim, P.P. Vaidyanathan and R. Ansari, "A new class of two-channel biothogonal filter banks and wavelet bases," *IEEE Trans. SP.*, vol. 43, pp. 649-664, Mar. 1995.

[6] S. Srirangarathan, D. R. Bull, and D. W. Redmill, "The design of low complexity two-channel lattice-structure perfect-reconstruction filter banks using genetic algorithms," in *Proc. IEEE ISCAS'97*, vol. 4, pp. 2393 –2396, Jun. 1997.

[7] W. Liu, S. C. Chan, and K. L. Ho, "Multiplier-less Low-delay FIR and IIR Wavelet Filter Banks with SOPOT Coefficients," in *Proc. IEEE ICASSP'2000, Istanbul, Turkey*, May 2000.

[8] I. Daubechies and W. Sweldens, "Factoring wavelet transform into lifting steps," *J. Fourier Anal. Appl.*, vol. 4, no. 3, pp. 247-269, 1998.

[9] A. G. Dempster and M. D. Macleod, "Use of minimum-adder multiplier blocks in FIR digital filters," *IEEE Trans. CAS. II*, vol. 42, pp. 569-577, Sept. 1995.

[10] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-time Signal Processing*. 2nd Edition, NJ, Prentice Hall.

[11] S. C. Chan, J. S. Mao and K. L. Ho, "A new design method for two-channel perfect reconstruction IIR filter banks," *IEEE Signal Processing Letters*, vol. 7, pp. 221-223, Aug. 2000

	$H_0(z)$ -39.084dB, avg. SOPOT term = 2.38 $\beta(z)$ (nonlinear FIR filter)			$H_1(z)$ -39.48dB, avg. SOPOT term = 2.13 $\alpha(z)$ (nonlinear FIR filter)		
n		PWL	Reg. (d_{n+1})		PWL	Reg. (d_{n+1})
0	$2^{-5}+2^{-7}$	<1|14>	<1|14>	-2^{-7}	<2|18>	<2|18>
1	$-2^{-3}+2^{-6}$ -2^{-8}	<1|15>	<1|15>	$2^{-5}-2^{-7}$	<2|18>	<2|18>
2	$2^{-1}+2^{-5}$ $+2^{-8}$	<1|15>	<1|15>	$-2^{-4}+2^{-6}$ -2^{-8}	<2|17>	<2|18>
3	$2^{0}-2^{-2}$ -2^{-6}	<1|13>	<2|15>	$2^{-4}+2^{-6}$ $+2^{-8}$	<2|17>	<2|18>
4	$-2^{-2}-2^{-4}$ $+2^{-7}$	<1|14>	<2|15>	$-2^{-2}+2^{-4}$ $+2^{-7}$	<2|17>	<2|18>
5	$2^{-2}-2^{-5}$ -2^{-7}	<1|14>	<2|15>	$2^{-1}+2^{-3}$ -2^{-6}	<2|18>	<3|18>
6	$-2^{-3}-2^{-5}$	<1|12>	<3|15>	$2^{-1}+2^{-3}$ $+2^{-5}$	<2|18>	<4|18>
7	$2^{-3}-2^{-7}$	<1|14>	<3|15>	$-2^{-2}+2^{-5}$	<2|18>	<4|18>
8	$-2^{-3}+2^{-5}$ $+2^{-8}$	<1|15>	<3|15>	$2^{-3}-2^{-9}$	<2|17>	<4|18>
9	2^{-4}	<1|11>	<3|15>	$-2^{-4}-2^{-6}$	<2|17>	<4|18>
10	$-2^{-5}-2^{-6}$ $+2^{-8}$	<1|15>	<3|15>	$2^{-4}-2^{-6}$ $+2^{-9}$	<2|17>	<4|18>
11	$2^{-5}-2^{-7}$	<1|14>	<3|15>	$-2^{-5}+2^{-8}$	<2|17>	<4|18>
12	-2^{-6}	<1|13>	<3|15>	2^{-6}	<2|19>	<4|19>
13				-2^{-7}	<2|17>	<4|19>
14				2^{-8}	<2|17>	<4|19>

Design Results	
Overflow Possibility	Zero
Input Format	<1|7>
Output Accuracy (fractional side)	-96.6dB (accuracy > 16-bit)
Output Wordlength	23-bit
Number of Adders in the MB	
$\beta(z)$	$\alpha(z)$
10	9
Esimated number of adder cells (with fixed wordlength of 24-bit using MBs)	1104
Estimated number of adder cells (with optimized wordlength using MBs)	825 (saved 27%)

Table 1. Filter banks results of example 5.1. PWL : product word length. Reg. : register.

	$H_0(z)$ -51.809dB, avg. term = 3.36	$H_1(z)$ -51.25dB, avg. term = 3.25	
n	$\beta(z)$ (numerator direct form)	n	$\alpha(z)$ (Linear Phase FIR Filter)
0	$-2^{-6}+2^{-9}+2^{-11}$	0,15	$-2^{-8}+2^{-10}$
1	$2^{-6}-2^{-10}$	1,14	$2^{-7}+2^{-12}$
2	$-2^{-4}+2^{-8}+2^{-10}$	2,13	$-2^{-6}-2^{-9}+2^{-11}-2^{-15}$
3	$2^{-1}-2^{-3}-2^{-6}+2^{-8}-2^{-10}$	3,12	$2^{-5}+2^{-9}$
4	$2^{1}-2^{-1}+2^{-3}-2^{-6}+2^{-9}$	4,11	$-2^{-4}+2^{-8}+2^{-10}-2^{-13}$
5	$2^{1}-2^{-2}-2^{-3}+2^{-5}+2^{-10}$	5,10	$2^{-3}-2^{-5}+2^{-7}+2^{-12}$
6	$2^{-1}+2^{-3}-2^{-5}-2^{-8}$	6,9	$-2^{-2}+2^{-4}-2^{-7}-2^{-11}$
n	$\beta(z)$ (denominator with lattice coefficients)	7,8	$2^{-1}+2^{-3}+2^{-7}-2^{-9}$
0	2^{-0}		
1	$2^{-0}-2^{-4}+2^{-8}+2^{-9}-2^{-11}$		
2	$2^{-0}-2^{-2}+2^{-5}+2^{-10}$		
3	$2^{-2}-2^{-8}-2^{-11}$		
4	$-2^{-5}-2^{-7}$		
5	$2^{-7}+2^{-8}-2^{-10}$		
6	$-2^{-9}-2^{-11}$		

Table 2. Filter banks results of example 5.2.

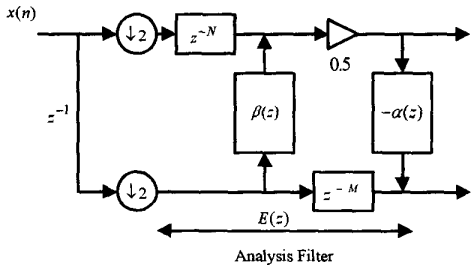

Fig.1. The biothogonal filter banks (analysis filter).

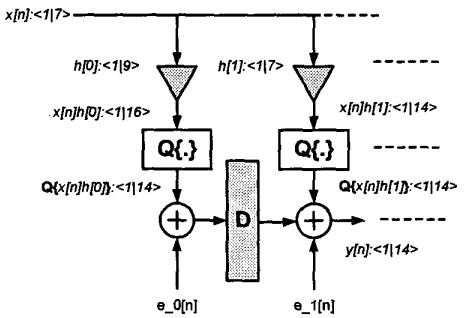

Fig. 2. Typical digital FIR filters with round-off noise model.

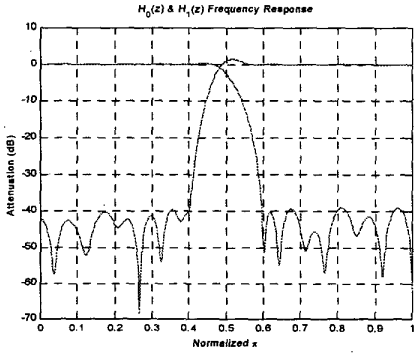

Fig. 3. Frequency responses of the two-channel FB in example 5.1.

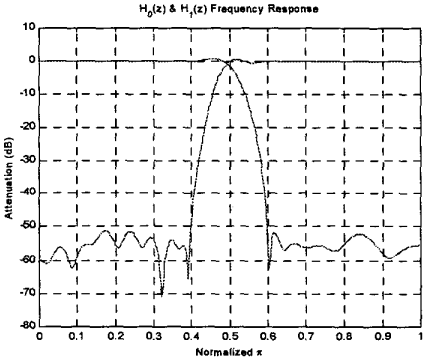

Fig. 4. Frequency responses of the two-channel FB in example 5.2.