World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:1, No:6, 2007

A Finite Precision Block Floating Point Treatment to Direct Form, Cascaded and Parallel FIR Digital Filters

Abhijit Mitra

Abstract—This paper proposes an efficient finite precision block floating point (BFP) treatment to the fixed coefficient finite impulse response (FIR) digital filter. The treatment includes effective implementation of all the three forms of the conventional FIR filters, namely, direct form, cascaded and parallel, and a roundoff error analysis of them in the BFP format. An effective block formatting algorithm together with an adaptive scaling factor is proposed to make the realizations more simple from hardware view point. To this end, a generic relation between the tap weight vector length and the input block length is deduced. The implementation scheme also emphasises on a simple block exponent update technique to prevent overflow even during the block to block transition phase. The roundoff noise is also investigated along the analogous lines, taking into consideration these implementational issues. The simulation results show that the BFP roundoff errors depend on the signal level almost in the same way as floating point roundoff noise, resulting in approximately constant signal to noise ratio over a relatively large dynamic range.

Keywords— Finite impulse response digital filters, Cascade structure, Parallel structure, Block floating point arithmetic, Roundoff error.

I. Introduction

FINITE impulse respone (FIR) digital filters, in general, exhibit certain desirable characteristics those are needed to ensure proper implementation in hardware. These include stability and realizability with appropriate finite delays. Among the other advantages of FIR filters, a few mentionable ones are the easeness to design exactly linear phase filters which is very useful for speech processing and data transmission and the characterization of finite precision roundoff error which can be made small enough for nonrecursive realizations. Although FIR filters also suffer from limitations like the impulse response duration must adequately approximate sharp cut-off filters, which, in turn, implies a large amount of processing, nevertheless, for its numerous advantages, such filters have been frequently utilized in many signal processing applications since the last five decades. An alphabetical list of many important works among these is given in the references [1]-[46]. Some of these studies have dealt with implementational issues [1], [5], [15], [19], [20], [21] mainly concerning with different types of FIR structures, while some other have attempeted to design optimal FIR structures [24]-[27] using different techniques such as integer programming, block-Z transform or over a discrete powers-of-two coefficient space. A few works, reported in the literature, focus on the performance analysis of FIR filters when realized in finite precision arithmetic with two widely known data formats, namely, fixed-point (FxP) and floating-point (FP) representation systems, by investigating the associated quantization errors [1], [6]-[8], [13], [18], [23], [28], [37]. In all these finite precision studies about FIR filters, however, it is found out that incorporating FxP format in any implmentational scheme gives the provision to take advantages like less computational complexity and power comsumption. However, the main drawback of this arithmetic is the limited dynamic range. Such a problem can be eluded using FP format by paying the price of increased hardware complexity. In a few other studies, efforts have been made to carry out the filter implementation with a viable alternative of normalized FP concept, called, block-floating-point (BFP) representation [41], [38] where the incoming data is partitioned into non-overlapping blocks and depending on the relative magnitudes of the data samples in each block, a common exponent is assigned. Thus the usual filtering computations under this arithemtic can be carried out in a FxP like manner while the common exponent provides the required wider dynamic range. To capitalize these benefits by employing the aforesaid BFP format, some studies have concentrated on the roundoff error properties of the same [2], [22], [30]-[33] in order to find out more deterministic error bounds for such realizations. However, to the best of our knowledge, no work has so far been reported in the literature with BFP treatment to direct, cascade and parellel forms of FIR filter together with a deterministic view proposing a detailed operation that include
(a) investigating the mutual relationship of filter length and input block length,
(b) an optimum adaptive scaling factor for BFP formatting to prevent overflow,
(c) an exponent update mechnism for implementational easeness, and,
(d) finding the lowest possible combinational error that emerges from finite precision analysis.

In this paper, we propose efficient realizations of fixed coefficient direct form, cascaded and parallel FIR digital filters employing the BFP arithmetic considering all the aforesaid points so that the treatment becomes deterministic in terms of implementation and error bound. Towards this end, a lemma has been proposed which does not restrict the block length to be equal to the filter length only but permits it to be longer as well. To update the exponent(s) upon arrival of new data sample(s), a block-by-block updating strategy is adopted to make the suggested scheme more attractive from the practical

Manuscript received Janaury 2, 2006.

A. Mitra is with the Department of Electronics and Communication Engineering, Indian Institute of Technology (IIT) Guwahati, North Guwahati - 781039, India. E-mail: a.mitra@iitg.ernet.in.

implementation point of view. An adaptive scaling factor is suggested to prevent overflow in the filtering process when the mantissas are taken either from a single block or from two adjacent blocks during the interblock transition. Additionally, the behavior of associated roundoff noise is investigated with appropriate finite precision BFP formats for the filter coefficient vector as well as the input data vector and it is observed that the simulation studies of the proposed realization have confirmed acceptable SNR characteristics over a wide dynamic range.

The paper is organized as follows. Section II briefly discusses about the BFP arithmetic fundamentals. The proposed implementation of direct form FIR filter with BFP arithmetic is dealt with in details in Section III and the similar idea is extended for cascaded and parallel realizations in Section IV. Section V gives a detailed analysis of roundoff noise in finite precision for all the three realizations including several error bounds. The paper is concluded by summarizing the important concepts introduced here in Section VI and with another new technique, where active investigation is still going on, in Appendix.

II. The BFP Arithmetic Fundamentals

We briefly describe the necessary background material first. In [31], a BFP arithmetic and the finite wordlength properties of the same are studied at length. It has been stated there that the BFP representation can be considered as a special case of FP format, where the incoming data are grouped into nonoverlapping blocks of N consecutive samples and each block has a joint scaling factor corresponding to the data samples with the largest magnitude in the block. In other words, given a block vector $\mathbf{x} = [x_1, ..., x_N]$, it can be represented as $\mathbf{x} = [\overline{x}_1, ..., \overline{x}_N].2^{\gamma} = \overline{\mathbf{x}}.2^{\gamma}$ where $\overline{x}_k (= x_k.2^{-\gamma})$ represent the mantissas for $k = 1, 2, ..., N$ and the block exponent γ is defined as

$$\gamma = \lfloor log_2 Max \rfloor + 1 + S \tag{1}$$

where $Max = max(|x_1|, ..., |x_N|)$, '$\lfloor . \rfloor$' is the so-called floor function, meaning rounding down to the closest integer and the integer S is a scaling factor needed to prevent the overflow during filtering operation. For the presence of S, the range of the mantissas are given as $|\overline{x}_k| \in [0, 2^{-S})$. Note that such a scaling term is not needed in simple BFP format data representation and thus the mantissa range would be $|\overline{x}_k| \in [0, 1)$ in that case. However, the scaling factor S can be calculated from the inner product computation during filtering operation. The filter output is usually expressed as an inner product in BFP format as

$$\begin{aligned} y(n) &= \langle \mathbf{w}, \mathbf{x}(n) \rangle = \mathbf{w}^T \mathbf{x}(n) \\ &= [w_0 \overline{x}(n) + \cdots + w_{L-1} \overline{x}(n - L + 1)].2^{\gamma} \\ &= \overline{y}(n).2^{\gamma} \end{aligned} \tag{2}$$

where $\mathbf{w}$ is the length L fixed coefficient vector of the direct form FIR filter and $\mathbf{x}(\mathbf{n})$ is the data vector at the nth index, represented in BFP format. For no overflow in $y(n)$, we require $|\overline{y}(n)| \leq 1$ at every time index, which can be satisfied by selecting [22]

$$S \geq S_{min} = \lceil log_2 (\sum_{k=0}^{L-1} |w_k|) \rceil \tag{3}$$

where '$\lceil . \rceil$' is the so-called ceiling function, meaning rounding up to the closest integer. Note that, if (B_d + one sign) bits are used to represent each mantissa within the block and if (B_γ + one sign) bits are used to account for the block exponent, then effectively, under BFP system, each sample can be equivalently represented with $(B_d + 1) + (B_\gamma + 1)/N$ bits because the block exponent is taken only once for the whole block. This particular strength makes this format more considerable than FxP or FP systems.

III. The Proposed Implementation of the Direct Form FIR Filter with BFP Arithmetic

In the suggested method, we first format the filter coefficient vector in the BFP representation as $\mathbf{w} = \overline{\mathbf{w}}.2^{\psi}$ where ψ is a block-exponent and is chosen so as to ensure that each $|\overline{w}_k| < 1$, $k = 0, 1, ..., L-1$. For such a choice of filter coefficient mantissa vector, eq. (2) is changed to

$$\begin{aligned} y(n) &= [\overline{w}_0 \overline{x}(n) + \cdots + \overline{w}_{L-1} \overline{x}(n - L + 1)].2^{\gamma+\psi} \\ &= \overline{y}(n).2^{\gamma+\psi} \end{aligned} \tag{4}$$

and eq. (3) takes the form of

$$S \geq S_{min} = \lceil log_2 L \rceil \tag{5}$$

as each $|\overline{w}_k| < 1$, $k = 0, 1, ..., L-1$.

We next partition the input data into non-overlapping blocks of N samples each and for any ith block ($i \in \mathbb{Z}$), the block exponent is assigned as

$$\gamma_i = ex_i + S_i \tag{6}$$

where

$$ex_i = \lfloor log_2 M_i \rfloor + 1 \tag{7}$$

and

$$M_i = max\{|x(iN)|, ..., |x(iN + N - 1)|\} \tag{8}$$

with $x(n)$ being the input sequence, $\mathbb{Z}$ denoting the set of integers and S_i being the proposed adaptive scaling factor. Having a common γ_i computed for any ith block, the block variables are expressed as

$$x(n) = \overline{x}(n).2^{\gamma_i}, \quad n \in (iN, ..., iN - N + 1), i \in \mathbb{Z}. \tag{9}$$

The above block separation and BFP formatting process is explained diagramatically in Fig. 1 with an example of $N = 4$.

Since the exponent γ_i is fixed for the block under consideration, calculations involving the block mantissas can be carried out in the usual fixed point like manner within a block but adjustments are necessary during the transition from one block to another and hence we need to track the change in the value of each γ_i in one separate register. For this purpose, a block

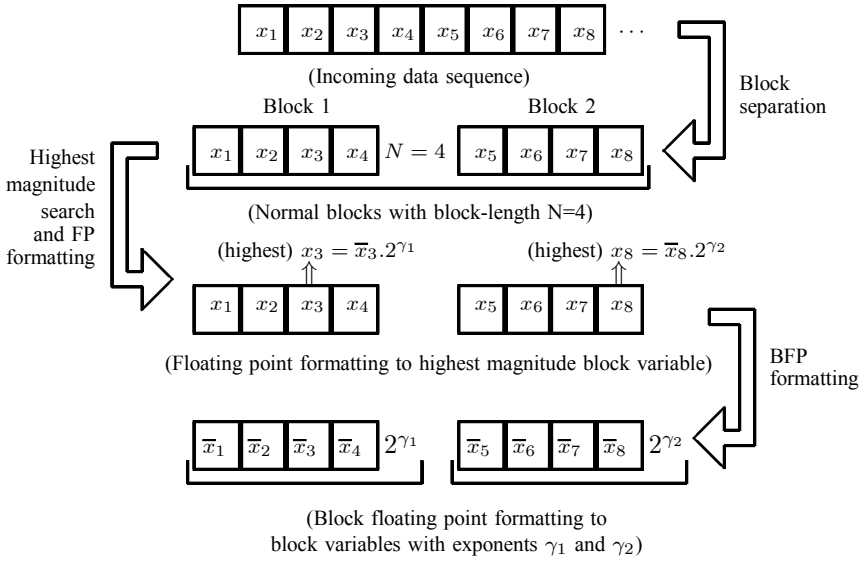

Fig. 1 Input data partitioning and BFP formatting mechanism for $N = 4$.

exponent update term $u_j(n)$ (jth term, at time instant n) is placed after each delay element, determined by the following equation

$$u_j(n) = \gamma_{\overline{x}(n-j)} - \gamma_{\overline{x}(n)} \tag{10}$$

where $\gamma_{\overline{x}(n-j)}$ and $\gamma_{\overline{x}(n)}$ represent the exponent values associated with $\overline{x}(n-j)$ and $\overline{x}(n)$ respectively at any time instant n. Fig. 2 shows the BFP implementation of a 3-tap FIR digital filter with two such update terms, $u_1(n)$ and $u_2(n)$. Here, the primed and unprimed entities symbolize the finite precision and infinite precision quantities respectively. From the above figure, it is easily understood that when the filtering operation involves data from only one block, the update terms assume zero value. On the other hand, during transition from one block to another, when the filter operates on segments of data from two adjacent blocks, the update terms need to be carefully determined so that no overflow occurs during the transition phase. As for example, consider the situation where $ex_i \geq \gamma_{i-1}$. Then, to avoid overflow within the ith block, we choose $\gamma_i = ex_i + S_{min}$. Next, consider filtering operation during the transition phase at an index $n = iN + k$, $k = 0, 1, ..., L-2$. This will involve the following data samples from the $(i-1)$th block: $[\overline{x}'(iN-L+k), ..., \overline{x}'(iN-1)]$ and rest from the ith block. To conform to the BFP format during the transition phase too, we employ rescaling of the above stated data segment from the $(i-1)$th block as per the following: $\overline{x}'(n) \rightarrow 2^{-(\gamma_i - \gamma_{i-1})}.\overline{x}'(n)$, where n denotes the relevant time indices in the $(i-1)$th block. Note that rescaling reduces the magnitudes of the samples within the domain of the filter and thus ensures that no overflow occurs either by the rescaling process or by the filtering operation during the block-to-block transition. For the case $ex_i \leq \gamma_{i-1}$ too, a similar treatment can be worked out. Thus we need to employ an adaptive scaling factor in the initial block formatting algorithm to have the non-positive update terms in any situation for overflow prevention. The combined approach is stated in the form of an algorithm below.

Algorithm: *Assign $S_{min} = \lceil log_2 L \rceil$ as the scaling factor to the initial data block and while considering about any general S_i for $i \geq 1$, assume $S_{i-1} \geq S_{min}$ and do the following:*
If $ex_i \geq ex_{i-1}$,
then assign $S_i = S_{min}$, s.t. $\gamma_i = ex_i + S_{min}$
else (i.e., $ex_i < ex_{i-1}$)
assign $S_i = (ex_{i-1} - ex_i + S_{min})$, s.t. $\gamma_i = ex_{i-1} + S_{min}$.

Note that when $ex_i \geq ex_{i-1}$, we can either have $ex_i + S_{min} \geq \gamma_{i-1}$ (Case A) implying $\gamma_i \geq \gamma_{i-1}$, or, $ex_i + S_{min} < \gamma_{i-1}$ (Case B) meaning $\gamma_i < \gamma_{i-1}$. However, for $ex_i < ex_{i-1}$ (Case C), we always have $\gamma_i \leq \gamma_{i-1}$. The above algorithm leads to the following theorem.

Theorem 1: The exponent update term will always be non-positive or zero preventing the possibility of overflow in the filtering operation during the transition from $(i-1)$th block to ith block as well as within the ith block, if

$$S_i = max\{\lceil log_2 L \rceil, (\gamma_{i-1} - ex_i)\}. \tag{11}$$

Proof: The proof is trivial from all the three conditions of the above algorithm. ■

Next we need a detailed study to find out a relation between the input block length N and the filter length L in order to investigate the implementational easeness that comes out from the correlation of these two, if any. This is stated in the form of the lemma given below.

Lemma 1: At any time index n, at the most $(j+2)$ adjacent blocks are involved in the filtering process, if

$$\lfloor \frac{L-2}{j+1} + 1 \rfloor \leq N \leq \lfloor \frac{L-2}{j} \rfloor \quad \forall j \in \mathbb{Z}_p \tag{12}$$

where $\mathbb{Z}_p$ is the set of all positive integers except zero.

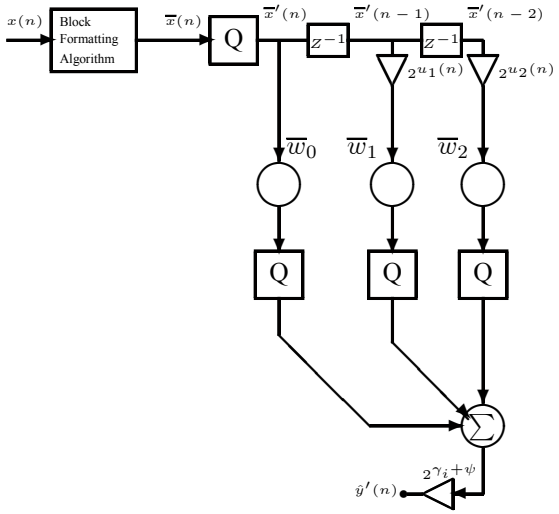

Fig. 2 Block floating point implementation of a 3-tap FIR digital filter, where 'Q' indicates the quantization operation.

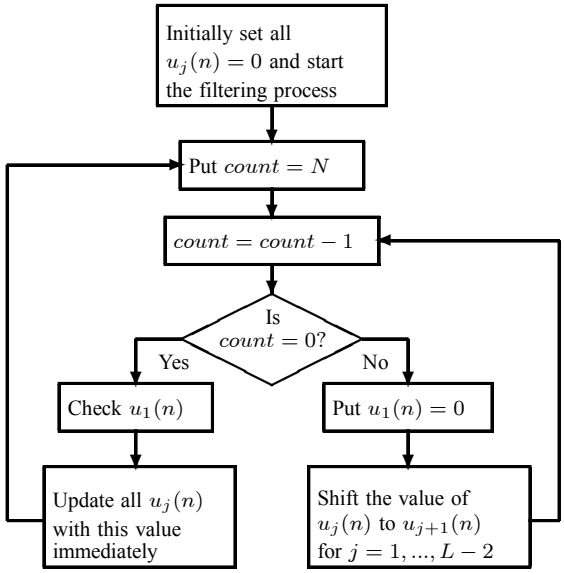

Fig. 3 A flow chart for efficient block exponent update technique.

Proof: Consider the nth time instant when the filter output

$$y(n) = x_k(n)w_0 + \langle \mathbf{w}_f, \mathbf{x}_{i-1} \rangle + x_{k-L+1}(n)w_{L-1} \quad (13)$$

where the vector $\mathbf{x}_{i-1} = [x_{k-1}(n), \cdots, x_{k-L+2}(n)]$ represents any $(i-1)$th block, $\mathbf{w}_f = [w_1, \cdots, w_{L-2}]$ and the data samples $x_k(n)$ and $x_{k-L+1}(n)$ are within the ith and $(i-2)$th blocks respectively. Quite clearly, the above filtering index incorporates 3 consecutive blocks when the relation $N \leq (L-2)$ is true and the same continues for $N \geq \lfloor \frac{L}{2} \rfloor$. Next, consider another mth time index with the output

$$\begin{aligned} y(m) &= x_k(m)w_0 + \langle \mathbf{w}_{f1}, \mathbf{x}_{i-1} \rangle + \langle \mathbf{w}_{f2}, \mathbf{x}_{i-2} \rangle \\ &\quad + x_{k-L+1}(m)w_{L-1} \end{aligned} \quad (14)$$

where the vector $\mathbf{x}_{i-1} = [x_{k-1}(m), \cdots, x_{k-\frac{L}{2}+1}(m)]$ represnts $(i-1)$th block, the $(i-2)$th block vector $\mathbf{x}_{i-2} = [x_{k-\frac{L}{2}}(m), \cdots, x_{k-L+2}(m)]$, $\mathbf{w}_{f1} = [w_1, \cdots, w_{\frac{L}{2}-1}]$, $\mathbf{w}_{f2} = [w_{\frac{L}{2}}, \cdots, w_{L-2}]$, and $x_k(n)$ and $x_{k-L+1}(n)$ are within the ith and $(i-3)$th blocks respectively (with the assumption of $(L-2)$ as even). The above equation then holds true for the integer range $\lfloor \frac{L-2}{3} + 1 \rfloor \leq N \leq \lfloor \frac{L-2}{2} \rfloor$ and the filtering process can be further subdivided in the same way with shorter block lengths. Thus follows the generalized statement of the above lemma. ■

Corollary 1: When the input sequence block length is greater than or equal to the filter weight length minus one, i.e., $N + 1 \geq L$, the filtering operation involves at the most two blocks at any time index p.

Proof: Putting $j = 0$ on the left hand side non-equality of eq. (12) and noticing the fact that there exist some time indices when the filtering operation must involve two blocks, however lengthy a block may be (except for the case $N \to \infty$ which is equivalent to fixed point case), we get the above corollary. ■

Having a generalized relation between the filter length and block length, it would now be appopriate to investigate a suitable realization scheme of exponenet update terms. To elude the evaluation of the update term after each time instant, we now propose an efficient exponent updating technique which updates all the $u_j(n)$ by actually updating only $u_1(n)$ after a periodic interval, provided the block length is chosen as stated in the above corollary. Usually, we assume a block length greater than the filter coefficient length to take the longer effects of the fixed point operational advantages for intra-block filtering. Utilizing such a relationship between N and L, it can easily be understood from Fig. 2 that whenever $u_1(n)$ assumes a nonpositive value, the value should be immediately transferred to all the other update registers and as long as $u_1(n) = 0$ (it will remain zero for $N-1$ time instants after taking a nonzero value), the null value has to be propagated to other update registers by sequential delay elements, eliminating the need of update operation after each time instant. A flow chart is given in Fig. 3 to implement such a mechanism. However, the situation is not so simple in the cases that involve more than two blocks for operation at a time and hence require more complicated exponent update mechanism.

Considering the above discussed implementational issues, the section can then be concluded with a general comment on the suitability of such BFP realizations in the form of the following theorem and, towards the end of this paper, as a corollary.

Theorem 2: Among all the digital filter structures, most suitable to BFP realization are those which are *canonic* with respect to delays.

Proof: A digital filter structure is said to be *canonic* if the number of delays in the filter structure is equal to the order of the transfer function. From block variable formation

view point in BFP arithmetic, this is desirable to form the same only once for effective implementation. A class of the equivalent structures of infinite impulse response (IIR) filters provide this canonic property and therefore, it also offers the same advantage when implemented using finite precision BFP arithmetic. On the other hand, both direct and transposed form FIR filters are canonic with respect to delays and provide a straightforward BFP realization. Thus follows the above theorem considering the realization issues. ■

Note that from quantization noise consideration, both the classes of FIR filters offer a downright BFP quantization noise treatment (discussed in Section V) with almost identical finite precision behavior. However, analyzing the canonic class IIR filters often becomes tedious for two basic reasons. Firstly, it is difficult to search a structure having the least quantization effects in finite precision. Secondly, investigating block variables as intermediate data functions is more hard than to do the same with primary variables as in FIR case. At the end of Section V, a generalization of this effect is presented in the form of a corollary.

IV. Extension to the Cascaded and Parallel Realizations

So far we have only considered the efficient implementation including how to prevent the possibility of overflow during the filtering operation taking into account the intra-block and inter-block data samples for direct form FIR filters. In this section, we extend our treatment to cascade and parallel structures of direct form FIR filters.

A. Cascade Structure

If M different direct form FIR sections are adjoined in cascade fashion with respective transfer function (TF) $H_p(z)$, $p = 1, 2, ..., M$, the resultant TF for such a cascaded structure comes as

$$H_{eq}(z) = \prod_{p=1}^{M} H_p(z) \tag{15}$$

where each individual sections are usually chosen as second order sections (to realize a complex zero with real filter coefficients) or as first order sections for simplicity. One possible difficulty that arises with cascade form is deciding about pole-zero pairing for IIR filters. However, for FIR filtering, such a case is not of our interest. Also, in cascade structure, no overflow will occur if all the individual sections ensure $|\overline{y}(n)| \leq 1$ at every time index, which is already shown in the proposed direct form realization. Nevertheless, a finite precision noise usually comes in this structure modeling the independent noise sources as additive noise sources, which is discussed in Section V.

B. Parallel Structure

For M different direct form sections with the same TFs as mentioned above, the resultant TF with parallel realization can be written as

$$H'_{eq}(z) = \sum_{p=1}^{M} H_p(z). \tag{16}$$

Here, unlike the cascade form, we need to add up all the outputs, which, in turn, necessitates the structure to scale all the input sections by $\frac{1}{M}$ in order to prevent overflow at the final stage. The roundoff error analysis of such a structure, however, is easier in comparison with cascade structures.

In the next Section, we would deal with the finite precision roundoff noise investigation of all these three different structures, starting with the direct form and then gradually extending the treatment to cascade and parallel structures.

V. A Quantization Error Analysis

A. Rounoff Error Model

The direct application of the roundoff error models used with FxP and FP format is not possible for the case of BFP representation. The additive roundoff error model of FxP system can not be used as BFP is a scaled number representation system with distinct block exponents for different blocks. Again, the relative roundoff error model of FP arithmetic can not be utilized because BFP format mantissas are not normalized. Therefore, the BFP quantization error is modeled with a scaled additive roundoff error model, defined as follows

$$\begin{aligned} \alpha &= Q[x_i] - x_i \\ &= (Q[\overline{x}_i] - \overline{x}_i).2^{\gamma_i} \\ &= e_m.2^{\gamma_i} \end{aligned} \tag{17}$$

where $Q[.]$ denotes the quantized value of a quantity and e_m is the mantissa quantization error which can be assumed to be an uncorrelated random variable. The block exponents γ_i are also assumed to be uncorrelated. If rounding-to-nearest is used as the rounding method, the roundoff error α has zero mean and variance

$$\sigma_\alpha^2 = \sigma_{e_m}^2.E[2^{2\gamma_i}] = \frac{2^{-2B_d}}{12}.\sum_{l=1}^{N_\gamma} p_\gamma(\gamma_l)2^{2\gamma_l} \tag{18}$$

where (one sign + B_d) bits have been used to represent each datum mantissa, $p_\gamma(\gamma_l)[l = 1, ..., N_\gamma]$ is the *probability mass function* of the block exponents and N_γ is the available distinct block exponent levels. Deduction of $p_\gamma(\gamma_l)$ is the most tedious job in the roundoff error model and thus it is approximated using some marginal distributions which usually leads to good results. If we assume that the signal is Gaussian distributed (i.i.d.) with variance σ_u^2, then the distribution of block exponents becomes

$$p_\gamma(\gamma_l) = [erf(\frac{2^{\gamma_l - S_{min}}}{\sqrt{2}\sigma_u})]^N - [erf(\frac{\frac{1}{2}2^{\gamma_l - S_{min}}}{\sqrt{2}\sigma_u})]^N \tag{19}$$

with $erf(x)$ being the *error function*, i.e.,

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt. \tag{20}$$

B. Roundoff Error Calculation of the Direct Form FIR Filter Structure

We start the roundoff noise analysis with direct form filters. From Fig. 2, it can be easily understood that the roundoff error variance in the output of a direct form FIR digital filter is contributed by three basic quantization processes [6], namely,
- the input data sample(s) quantization, better known as A-D conversion noise,
- the quantization of the filter coefficients, and,
- the uncorrelated error due to rounding of arithmetic operations in calculating the filter ouput.

These three types of noise, in connection with the proposed implementations, are discussed below. Note that the correlated roundoff noise, known as limit cycles [36], mainly concerns with IIR filters and therefore is beyond the scope of our consideration.

B.1 A-D Conversion Noise:

The roundoff noise variance in the output due to kth input quantization point is

$$\sigma_{i,k}^2 = |w_k|^2 \underbrace{\sigma_{e_m}^2 . E(2^{2\gamma_i})}_{} = |w_k|^2 \sigma_\alpha^2 \tag{21}$$

where σ_α^2 refers to the variance of roundoff error α as indicated in eq. (18). As the different roundoff error sources are assumed to be uncorrelated, the total output roundoff error variance for any arbitrary phase direct form digital filter, due to input data quantization, becomes

$$\sigma_{i,total}^2 = \sigma_\alpha^2 \sum_{k=0}^{L-1} |w_k|^2. \tag{22}$$

For the linear phase case, $w_k = w_{L-1-k}$, for $k \in [0, L-1]$ and a marginally different version of the direct form with fewer multipliers, can be derived. Assuming L even in this case, eq. (22) changes to

$$\sigma_{i,total}^2 = \sigma_\alpha^2 \sum_{k=0}^{\frac{L}{2}-1} |w_k|^2. \tag{23}$$

Performances of A-D conversion noise in BFP format along with FP and FxP formats have been studied by quantizing uncorrelated Gaussian data to all these number representation systems, using same number of mantissa bits per sample for all of these three formats. The SNRs are plotted in Fig. 4 as a function of the input signal level. From the theory of BFP SNR calculation, we got approaximately 45.7 dB (for N=8) and 43.8 dB (for N=16), which have shown a very good agreement with the simulated results. Fig. 4 also indicates that when the SNR of FxP system decreases with the reduction in signal level, the SNRs of BFP and FP system remain almost constant over a large dynamic range. Additionally, with a small block length, the SNR of BFP format almost reaches the SNR of FP representation system.

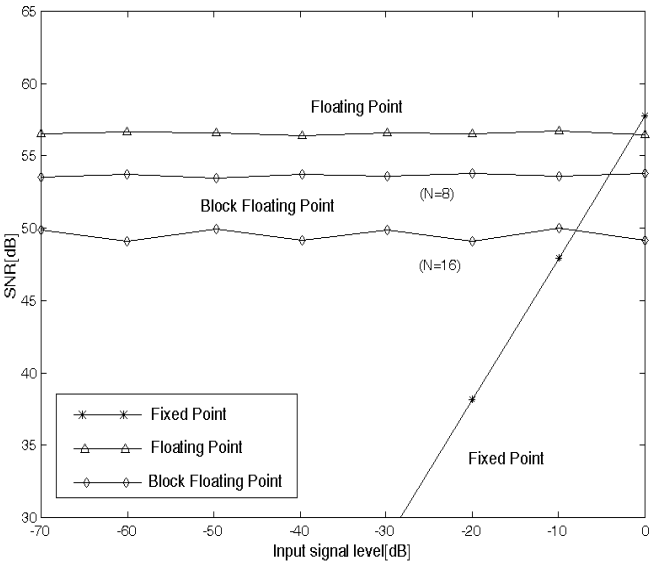

Fig. 4 SNR diagram for quantizing the uncorrelated Gaussian data with 1+7 bit FxP, 1+7+(1+3) bit FP and 1+7+(1+3)/N bit BFP format, with N=8 and N=16.

B.2 Filter Coefficient Quantization Noise:

For any arbitrary phase direct form filter, the total error due to filter quantization comes as

$$\sigma_{c,total}^2 = L\frac{2^{-2B_c}}{12} \tag{24}$$

using the independence assumption, where (one sign + B_c) bits are used to represent each coefficient. For the linear phase direct form filters, assuming L even, the above equation (24) changes to $\sigma_{c,total}^2 = L\frac{2^{-2B_c}}{24}$. A similar kind of bound is obtained by calculating the total coefficient roundoff error variance in the frequency domain as

$$\begin{aligned}\Psi_{c,total}^2(e^{j\omega}) &= \frac{2^{-2B_c}}{12}\sum_{k=0}^{\frac{L}{2}-1} 4cos^2[(\frac{L-1}{2}-k)\omega] \\ &= \frac{2^{-2B_c}}{12}[4cos^2\frac{\omega}{2} + \sum_{k=1}^{\frac{L}{2}-1} 4cos^2(k+\frac{1}{2})\omega].\end{aligned} \tag{25}$$

An example of filter coefficient quantization effect with 1+7+(1+3)/20 bit BFP format for a length 20 FIR equiripple low pass filter is shown in Fig. 5, which clearly depicts that the quantization noise is well within the acceptable limit within and beyond passband.

B.3 Filtering Operation Roundoff Error:

Under this, there are three different types of errors [2] those are frequently meet with during the operations for any linear

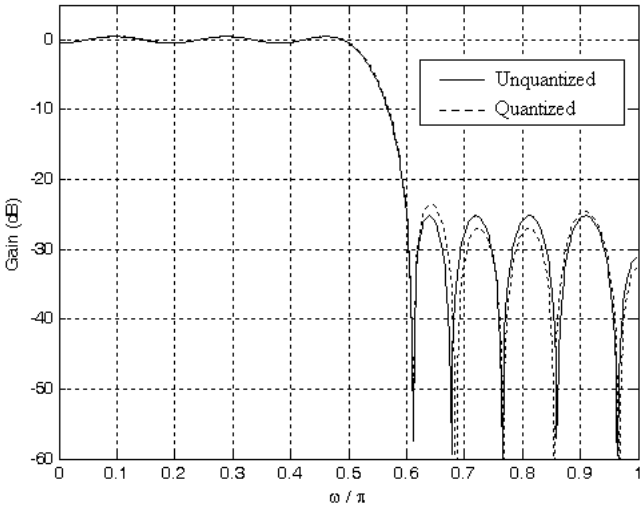

Fig. 5 Coefficient quantization effects on a length 20 direct form FIR equiripple low pass filter with 1+7+(1+3)/20 bit BFP format.

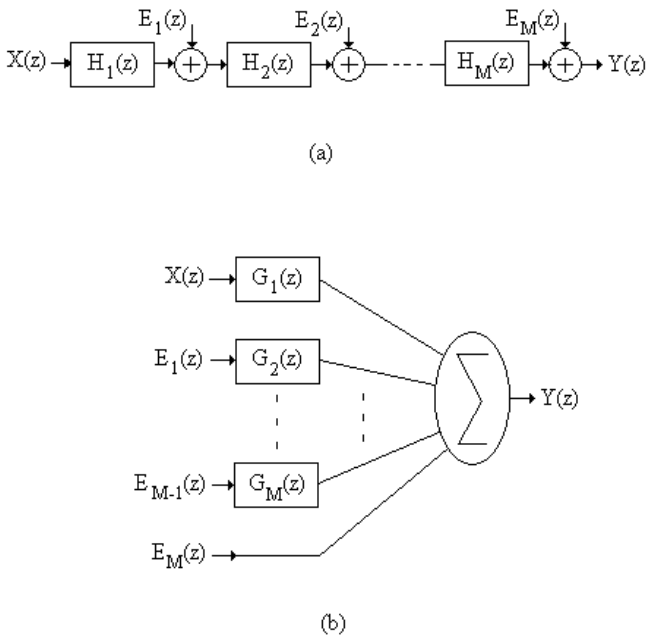

Fig. 6 Cascade structure in finite precision. (a) The cascade form with additive errors, and (b) the equivalent form.

system. These are as follows:
(a) *Block Denormalization Error*: We get this error, if, due to exponent increament, any mantissa register has to be right-shifed and the LSB is lost. Here, the total error can be expressed as

$$|\eta_{DN}| < 2^{-B_d+\gamma} \sum_{k=0}^{L-1} |w_k|. \quad (26)$$

Such an error occurs rarely in the case of FIR filters.
(b) *Multiplication Quantization Error*: This error occurs if quantization operation is done right after multiplication and is bounded by

$$|\eta_M| < L.2^{-B_d+\gamma}. \quad (27)$$

(c) *Addition Quantization Error*: This error comes into calculation by performing quantization right after addition/subtraction and is bounded by

$$|\eta_A| < 2^{-B_d+\gamma}. \quad (28)$$

We have preferred quantization after each multiplication and thus only error bound (b) has to be taken into account. Thus, in our case, the rounding operations after every multiplication in the process of calculating the filter output adds up to the total roundoff error variance

$$\sigma^2_{f,total} = E(2^{2\gamma_i}) \sum_{k=0}^{L-1} \sigma^2_f = L\frac{2^{-2B_d}}{12} E(2^{2\gamma_i}) \quad (29)$$

for any arbitrary phase direct form FIR filter and we get $\sigma^2_{f,total} = L\frac{2^{-2B_d}}{24}E(2^{2\gamma_i})$ for those of the linear phase (assuming L even).

Hence, the total output error variance comes as

$$\begin{aligned} \sigma^2_{total} &= \sigma^2_{i,total} + \sigma^2_{c,total} + \sigma^2_{f,total} \\ &= \sigma^2_{e_m}(\sum_{k=0}^{L-1} |w_k|^2 + L)E(2^{2\gamma_i}) + L\frac{2^{-2B_c}}{12} \quad (30) \end{aligned}$$

for an arbitrary phase direct form FIR filter and

$$\sigma^2_{total} = \sigma^2_{e_m}(\sum_{k=0}^{\frac{L}{2}-1} |w_k|^2 + \frac{L}{2})E(2^{2\gamma_i}) + L\frac{2^{-2B_c}}{24} \quad (31)$$

for a linear phase filter, assuming L even. The finite precision roundoff error investigation approach, presented in this subsection, can be extended to cascade and parallel structures with some additional considerations. This is discussed next.

C. Roundoff Error Analysis of Cascaded and Parallel Form

C.1 Cascaded Structure:

A finite precision cascaded structure with independent additive error models is shown in Fig. 6(a) and the equivalent parallel form of the same is shown in Fig. 6(b), where,

$$G_j(z) = \{\prod_{p=j}^{M} H_p(z) | j \in [1, M], j \in \mathbb{Z}\}. \quad (32)$$

For modeling the cascaded form in such a way, the overall error variance comes as

$$\sigma^2_{cas} = \sigma^2_{total}G^2_2(z) + \sigma^2_{e_1}G^2_2(z) + \sigma^2_{e_2}G^2_3(z) + ... + \sigma^2_{e_M} \quad (33)$$

where, any error variable $E_p(z) \leftrightarrow e_p(n)$ denotes the Z transform pair and $\sigma^2_{e_p}$ the corresponding variance. As before, σ^2_{total} denotes the quantity as presented in either eq. (30) or (31), depending upon the choice of phase of the direct form filters.

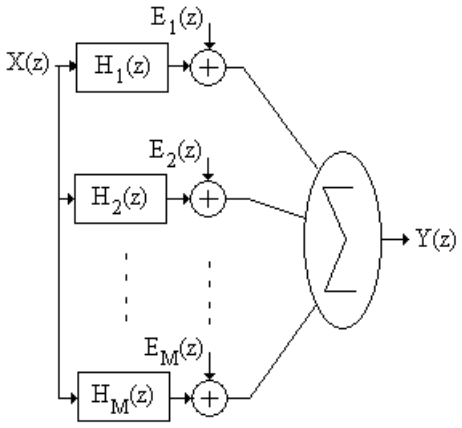

Fig. 7 Parallel structure in finite precision.

C.2 Parallel Structure:

Fig. 7 shows the finite precision parallel connection of the same direct form FIR sections with independent additive error models. Here, the error calculation is quite straightforward as it sums up all the individual error variences. However, for each section, here we have the error variance of $\sigma^{'2}_{total}$ instead of σ^2_{total} as each section is scaled *a-priori* by a scaling factor of $\frac{1}{M}$ in order to prevent overflow in the final output. In this case, the overall error variance therefore comes as

$$\sigma^2_{par} = M\sigma^{'2}_{total} + \sum_{p=1}^{M} \sigma^2_{e_p}. \tag{34}$$

Finally, a general comment can be made about the BFP realization of such digital filters in the form of the following corollary.

Corollary 2: Finite wordlength effects of FIR digital and/or adaptive filter structures in BFP arithmetic are easier to analyse, in comparison with IIR structures, due to their canonic property and non-correlated roundoff error behavior.

Proof: Combining Theorem 2, eq. (30) and (31), the above corollary can easily be observed. ■

VI. Conclusions

An efficient finite precision BFP realization of the fixed coefficient direct form FIR digital filter has been proposed and the approach has been extended to cascade and parallel structures. The proposed scheme enjoys higher flexibility in terms of the choice of block length which is not preconditioned to be equal to the filter length and a detailed analysis showing the mutual relationship of these two has also been carried out, keeping in mind the implementational simplicity, which, in turn, doesn't allow data samples to be taken from more than two blocks at any time instant. A block exponent update mechanism has been proposed where the block exponent is updated only once for each block, provided, the input sequence block length follows the above realtionship. An adaptive scaling factor has been suggested to prevent overflow in the filtering process when the mantissas are taken either from a single block or from two side-by-side blocks during the interblock transition. A roundoff error analysis has also been carried out for all the three structures. For this, appropriate finite precision BFP formats for the filter coefficient vector and the input data vector have been adopted and the filtering algorithm has been recast in terms of the chosen formats. The simulation results have also confirmed sufficient SNR over a large dynamic range. Currently, attempts are being made to extend this approach to the more challenging area of parallel interconnection of cascaded sections for improved performance at very high order for certain applications and is discussed in brief in the Appendix.

Appendix

Parallel Interconnection of Cascaded Subfilters

A recent paper [44] has proposed a class of IIR digital filter structures that make a compromise between standard cascade and parallel structures. It has been demonstrated there that in cases where the standard cascade and parallel forms are somewhat unusable, the proposed hybrid structure, termed as parallel interconnection of cascaded subfilters (PICS), performs well and in many applications like virtual reality, auralisation, musical instrument synthesis and high quality speech processing, where all-pole digital filters of order higher than 50 are needed, the characteristics of such hybrid PICS structure has been investigated with satisfactory results. The structure basically exploits the incomplete partial fraction expansion (IPFE) algorithm [14] to a list of poles in complex conjugate pairs so that it returns a polynomial in z^{-1} for each subfilter that is implemented as an FIR equaliser. In other words, given a rational polynomial $A(z) = u(z)/q(z)$, the IPFE algorithm finds the numerator polynomials $h(z)$ and $f(z)$ of a sum of two lower order rational functions

$$A(z) = \frac{h(z)}{r(z)} + \frac{f(z)}{s(z)} \tag{35}$$

where $r(z)s(z)$ is a given factorization of $q(z)$. The resultant PICS structures show lower roundoff noise levels than the actual cascade forms, and are more accurately synthesised than the parallel forms.

Our focus has been to extend this notion for the case of FIR filters in order to enjoy the above mentioned advantages with such a PICS form, which would consist of all first order FIR sections in all the parallel paths and each branch would be cascaded with k number of first order FIR sections to implement $h(k)z^{-k}$, provided $h(k)$ is factorizable k times with common factors. In particular, if $h(k) = h_1.h_2 \cdots .h_k$, any FIR order can be realized in this PICS form with simple first order sections only. Further, other ways are also possible to factorize any $h(k)$ with less number of common factors and active investigation on the same is now being carried out by the same author. As the simulation results are in preliminary stage, they are thus not shown here.

REFERENCES

[1] C. W. Barnes and S. Shinnaka, "Finite Word Effects in Block-state Realizations of Fixed Point Digital Filters," *IEEE Trans. Circuits Syst.*, vol. CAS-27, pp. 345-349, May 1980.

[2] P. H. Bauer, "Absolute Error Bounds for Block-Floating-Point Direct-Form Digital Filters," *IEEE Trans. Signal Processing*, vol. 43, no. 8, pp. 1994-1996, Aug. 1995.

[3] R. N. Bracewell, "The Fast Hartley Transform," *Proc. IEEE*, vol. 72, no. 8, pp. 1010-1018, Aug. 1984.

[4] C. Caraiscos and B. Liu, "A Roundoff Error Analysis of the LMS Adaptive Algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, no. 1, pp. 34-41, Feb. 1984.

[5] C. Caraiscos, "Implementation Issues in Digital Signal Processing," Ph.D. dissertation, Princeton University, Jan. 1984.

[6] D. S. K. Chan and L. R. Rabiner, "Analysis of Quantization Errors in the Direct Form for Finite Impulse Response Digital Filters," *IEEE Trans. Audio Electroaccoust.*, vol. AU-21, no. 4, pp. 354-366, Aug. 1973.

[7] D. S. K. Chan and L. R. Rabiner, "Theory of Roundoff Noise in Cascade Realizations of Finite Impulse Response Digital Filters," *Bell Syst. Tech. J.*, vol. 52, no. 3, pp. 329-345, Mar. 1973.

[8] D. S. K. Chan and L. R. Rabiner, "An Algorithm for Minimizing Round-off Noise in Cascade Realizations of Finite Impulse Response Digital Filters," *Bell Syst. Tech. J.*, vol. 52, no. 3, pp. 347-385, Mar. 1973.

[9] D. J. DeFatta, J. G. Lucas and W. S. Hodgkiss, *Digital Signal Processing: A System Design Approach*. New York: Wiley, 1990.

[10] P. M. Ebert, J. E. Mazo, and M. G. Taylor, "Overflow oscillations in digital filters," *Bell Sys. Tech. J.*, vol. 48, pp. 2999{3020, 1969.

[11] A. Erickson and B. Fagin, "Calculating FHT in Hardware," *IEEE Trans. Signal Processing*, vol. 40, no. 4, pp. 1341-1353, June 1992.

[12] B. Gold and C. M. Rader, *Digital Processing of Signals*. New York: Mcgraw-Hill, 1969.

[13] J. R. Heath, H. T. Nagle, Jr., and S. G. Shiva, "Realization of Digital Filters using Input-Scaled Floating-Point Arithmetic," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, no. 5, pp. 469-477, Oct. 1979.

[14] P. Henrici, *Partial Fractions*, volume 1 of "Applied and Computational Complex Analysis," Chapter 7. Wiley, 1974.

[15] O. Hermann and H. W. Schuessler, "On the Accuracy Problem in the Design of Nonrecursive Digital Filters," *Arch. Elek. Ubertragung*, vol. 24, pp. 525-526, 1970.

[16] L. B. Jackson, "Beginnings: The First Hardware Digital Filters," *IEEE Signal Proc. Magazine*, vol. 21, no. 6, pp. 55-81, Nov. 2004.

[17] L. B. Jackson, *Digital Filters and Signal Processing*, 3rd ed. Boston, MA: Kluwer, 1996.

[18] L. B. Jackson, "Roundoff-Noise Analysis for Fixed-Point Digital Filters Realized in Cascade or Parallel Form," *IEEE Trans. Audio Electroacoust.*, vol. AE-18, no. 2, pp. 107-122, June 1970.

[19] L. B. Jackson, J. F. Kaiser and H. S. McDonald, "An Approach to the Implementation of Digital Filters," *IEEE Trans. Audio Electroacoust.*, vol. AE-16, no. 3, pp. 413-421, Sept. 1968.

[20] T. Kailath, "A View of Three Decades of Linear Filtering Theory," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 146-181, March 1974.

[21] J. F. Kaiser, "Digital Filters," in *System Analysis by Digital Computer*, J. F. Kaiser and F. F. Kuo, Eds. New York: Wiley, 1966, pp. 218-285.

[22] K. Kalliojarvi and J. Astola, "Roundoff Errors in Block-Floating-Point Systems," *IEEE Trans. Signal Processing*, vol. 44, no. 4, pp. 783-790, April 1996.

[23] D. M. Kodek, "Performance Limit of Finite Wordlength FIR Digital Filters," *IEEE Trans. Signal Processing*, vol. 53, no. 7, pp. 2462-2469, July 2005.

[24] D. M. Kodek, "Design of Optimal Finite Wordlength FIR Digital Filters using Integer Programming Techniques," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, June 1980.

[25] W. Li and A. M. Peterson, "Block Z Transform and Its Application to FIR Filtering," *IEEE Trans. Signal Processing*, vol. 39, no. 10, pp. 2335-2343, Oct. 1991.

[26] Y. C. Lim and S. R. Parker, "Discrete coefficient FIR digital filter design based upon an LMS criteria," *IEEE Trans. Circuits Syst.*, vol. CAS-30, pp. 723-739, Oct. 1983.

[27] Y. C. Lim and S. R. Parker, "FIR filter design over a discrete powers of two coefficient space," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 583-590, June 1983.

[28] B. Liu and T. Kaneko, "Error Analysis of Digital Filters Realized in Floating-Point Arithmetic," *IEEE Proc.*, vol. 57, pp. 1735-1747, Oct. 1969.

[29] M. Martinez-Peiró *et. al.* (2002, February). FPGA Based FIR Filters using Distributed Arithmetic [Online]. Available: http://www.techonline.com/community/ed_resource/feature_article/20135.

[30] A. Mitra *et. al.*, "A Block Floating Point Treatment to the LMS Algorithm: Efficient Realization and Roundoff Error Analysis," *IEEE Trans. Signal Processing*, vol. 53, no. 12, pp. 4536-4544, Dec. 2005.

[31] A. Mitra, "On Finite Wordlength Properties of Block Floating Point Arithmetic," *Int. J. Signal Processing*, vol. 2, no. 2, pp. 120-125, July 2005.

[32] A. Mitra, "A New Block-based NLMS Algorithm and Its Realization in Block Floating Point Format," *Int. J. Info. Tech.*, vol. 1, no. 4, pp. 244-248, Dec. 2004.

[33] A. Mitra, "A New Way of Implementation and Associated Round-off Noise Analysis of Fixed Coefficient FIR Digital Filters Employing Block-Floating-Point Arithmetic," in *Proc. 3rd IEEE Benelux Signal Processing Symposium (SPS 2002)*, Leuven, Belgium, March 21-22, 2002, pp. 113-116.

[34] S. K. Mitra, *Digital Signal Processing: A Computer-based Approach*. New York: Mcgraw-Hill, 2001.

[35] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[36] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[37] A. V. Oppenheim and C. Weinstein, "Effects of finite register length in digital filtering and the fast Fourier transform," *Proc. IEEE*, vol. 60, pp. 957-976, Aug. 1972.

[38] A. V. Oppenheim, "Realization of digital filters using block floating point arithmetic," *IEEE Trans. Audio Electroacoust.*, vol. AE-18, no. 2, pp. 130-136, June 1970.

[39] L. R. Rabiner and B. Gold, *Theory and Applications of Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1976.

[40] L. R. Rabiner, "The Design of Finite Impulse Response Digital Filters using Linear Programming Techniques," *Bell Syst. Tech. J.*, vol. 21, pp. 1177-1198, July-Aug. 1972.

[41] K. R. Ralev and P. H. Bauer, "Realization of Block Floating Point Digital Filters and Application to Block Implementations," *IEEE Trans. Signal Processing*, vol. 47, no. 4, pp. 1076-1086, April 1999.

[42] S. Sridharan and G. Dickman, "Block floating point implementation of digital filters using the DSP56000," *Microprocess. Microsyst.*, vol. 12, no. 6, pp. 299-308, July-Aug. 1988.

[43] F. J. Taylor, "Block Floating Point Distributed Filters," *IEEE Trans. Circuits Syst.*, vol. CAS-31, pp. 300-304, Mar. 1984.

[44] M. Waters *et. al.*, "Parallel Interconnection of Cascaded Subfilters: Improved Performance at High Order," in *Proc. IEEE Int. Symp. Circuits, Syst. (ISCAS)*, Hong Kong, June 9-12, 1997, pp. 2216-2219.

[45] C. Weinstein and A. V. Oppenheim, "A Comparison of Roundoff Noise in Fixed Point and Floating Point Digital Filter Realizations," *Proc. IEEE*, vol. 57, pp. 1181-1183, Aug. 1969.

[46] C. Weinstein, "Quantization Effects in Frequency Sampling Filters," *NEREM Record*, 222, New York: Lewis Winner, 1968.

Abhijit Mitra was born in Serampore, India, in 1975. He received the B.E.(Honors) degree from Regional Engineering College, Durgapur, India, in 1997, the M.E.Tel.E. degree from Jadavpur University, India, in 1999 and the Ph.D. degree from Indian Institute of Technology, Kharagpur, India, in 2004, all in electronics and communication engineering. Since 2004, he has been with Indian Institute of Technology, Guwahati, India, as an Assistant Professor. His research interests include finite wordlength digital signal processing, statistical signal processing, adaptive signal processing and signal processing applications in wireless communications. He has over 30 research publications in these areas.

Dr. Mitra has been on the editorial board of International Journal of Signal Processing, International Journal of Information Technology and International Journal of Information Science since 2005. He has been a member of IEEE since 2003 and presently serves as a reviewer of IEEE Transactions on Signal Processing and IEEE Signal Processing Letters. He is the recipient of several national scholarships and research fellowships.