An Efficient Finite Precision Realization of the Block Adaptive Decision Feedback Equalizer

Rafi Ahamed Shaik* and Mrityunjoy Chakraborty†
*Indian Institute of Technology, Guwahati-781 039, India
Email: rafiahamed@iitg.ernet.in
†Indian Institute of Technology, Kharagpur-721 302, India
Email: mrityun@ece.iitkgp.ernet.in

***Abstract*— Recently, a block based adaptive decision feedback equalizer (ADFE) is presented which first uses an iterative scheme to evaluate a block of unknown decisions. FFT based block processing is then used on the received input block and the decision block to carry out the block ADFE operation. A direct floating point (FP) based realization of this scheme, however, pushes up the cost and complexity of processing hugely, as each FP operation involves several additional steps not present in its fixed point (FxP) counterpart. To overcome this problem, a block floating point (BFP) based treatment is presented in this paper for realization of the block ADFE. The proposed scheme, while maintaining FP like high dynamic range, deploys mostly FxP operations and thus reduces the processing cost and complexity substantially.**

I. Introduction

The adaptive decision feedback equalizer (ADFE) is an effective means for equalizing channels that exhibit spectral nulls and / or has a long impulse response (IR) giving rise to inter-symbol interference over a very large number of symbol periods. The linear equalizer is not a very effective option in such cases, due to the possibility of substantial noise enhancement and also due to a very large order requirement. The ADFE consists of a feed forward filter (FFF) and a feedback filter (FBF). The FFF, working directly on the received data, tries to equalize the anticausal part of the channel impulse response. The residual ISI at the FFF output is then canceled by passing the past decisions through an appropriately designed FBF and subtracting the FBF output from the FFF output. Both the FFF and the FBF coefficients are trained by some suitable adaptive algorithm. In this paper, we consider the simple LMS [8] based ADFE.

A common problem faced by the ADFE is that with increasing data transmission rate, the channel IR length increases and thus the order of both the FFF and the FBF increases. The resulting increase in complexity makes the real time operation of the ADFE difficult, specially in view of simultaneous shortening of the symbol period. Block processing [6] is one of the approaches to reduce complexities in digital filters, as the block based computations like convolutions and correlations can be implemented using FFT. However, the idea of block processing can not be applied directly to the ADFE, since, while block processing of the FFF input, i.e., received data is possible as these are known a priori, same is not true for the FBF input, i.e., decisions, which are unknown and are in fact sought to be evaluated by the ADFE. In [5], an iterative scheme is presented that evaluates the block of unknown decisions requiring only a few iteration steps. Computations within the iteration are block based and thus can be realized using FFT. Once the decisions are known, usual FFT based block processing techniques are applied to carry out the ADFE operation.

In a practical communication system, the input to the equalizer is in floating point (FP) form, caused by the need to amplify the weak, received signal with fluctuating signal level, by a programmable gain amplifier (PGA) that adjusts its gain continuously (by a power of two) for maximal utilization of the ADC dynamic range. A FP based processing, however, pushes up the cost and complexity of processing enormously, as computations in FP involve several additional steps not present in fixed point (FxP) computations. In this paper, we tackle this problem, by presenting a block floating point (BFP) treatment to the finite precision realization of [5]. In BFP, a common exponent is assigned to a block of data. As a result, computations involving these data require only simple FxP operations, while presence of the exponent maintains the desired high dynamic range. In recent years, the BFP format has been used extensively for efficient realization of various forms of digital and adaptive filters ([1]-[4]). The proposed treatment uses the philosophy of [3] and being based largely on FxP operations, achieves considerable speed up over a FP based realization of [5].

Throughout the paper, we follow the same notation as used in [5], namely, by $x(n)$ we denote a scalar quantity at time instant n, whereas by $\mathbf{x}_M(n)$, we denote either a $M \times 1$ filter coefficient vector at index n, or, a data vector $\equiv [x(n)\, x(n-1) \cdots x(n-M+1)]^t$ with $x(n)$ denoting the data input at the current index n. In addition $[\mathbf{x}]_{first.Q}$ and $[\mathbf{x}]_{last.Q}$ indicate respectively the first Q and last Q elements of the vector $\mathbf{x}$ and $X_{M,N}$ denotes a matrix of M rows and N columns. If the matrix is square, then the simpler notation X_M is used instead of $X_{M,M}$. The notation $\mathbf{X}_M$ is used to denote an M-length column vector in the frequency domain. Finally, characters with an overbar are used to indicate mantissas and Z_M for any integer , $M \geq 0$ denotes the set $\{0, 1, \cdots, M-1\}$.

II. Implementation of Block ADFE

Consider the block ADFE [5] that takes $x(n)$, $n \in Z$ as the input and updates the equalizer weights once per block of size Q. For the j-th block, the ADFE operation is given by the following block formulated set of equations:

$$\mathbf{y}_Q(jQ+Q-1)=X_{Q,M}\,\mathbf{w}_M^f(j)+D_{Q,L}\,\mathbf{w}_L^b(j), \quad (1)$$

$$\mathbf{d}_Q(jQ+Q-1)=f\{\mathbf{y}_Q(jQ+Q-1)\}, \quad (2)$$

$$\mathbf{e}_Q(jQ+Q-1)=\mathbf{d}_Q(jQ+Q-1)-\mathbf{y}_Q(jQ+Q-1) \quad (3)$$

$$\mathbf{w}_M^f(j+1)=\mathbf{w}_M^f(j)+\mu\,X_{Q,M}^H\,\mathbf{e}_Q(jQ+Q-1), \quad (4)$$

$$\mathbf{w}_L^b(j+1)=\mathbf{w}_L^b(j)+\mu\,D_{Q,L}^H\,\mathbf{e}_Q(jQ+Q-1). \quad (5)$$

Where,

$$X_{Q,M}=\begin{bmatrix} x(jQ+Q-1) & \cdots & x(jQ+Q-M) \\ \vdots & \ddots & \vdots \\ x(jQ) & \cdots & x(jQ-M+1) \end{bmatrix},$$

$$D_{Q,L}=[D_{Q,Q-1}^1 \;\; D_{Q,L-Q+1}^2],$$

with

$$D_{Q,Q-1}^1=\begin{bmatrix} d(jQ+Q-2) & \cdots & d(jQ) \\ \vdots & \ddots & \vdots \\ d(jQ-1) & \cdots & d(jQ-Q+1) \end{bmatrix},$$

and

$$D_{Q,L-Q+1}^2=\begin{bmatrix} d(jQ-1) & \ldots & d(jQ-L+Q-1) \\ \vdots & \ddots & \vdots \\ d(jQ-Q) & \ldots & d(jQ-L) \end{bmatrix}.$$

In (2), $f\{.\}$ is a Q-dimensional decision device in which the distance between its input and discrete output is minimum in the Euclidean sense. Also, the FBF order L is assumed above to be greater than Q (cases where $L \leq Q$ form a special case of the proposed treatment).

In the proposed scheme, the equalizer weight vector $\mathbf{w}(j) = [\mathbf{w}_M^{f\,t}(j) \;\; \mathbf{w}_L^{b\,t}(j)]^t$ is represented in a BFP format as :

$$\mathbf{w}(j)=[\overline{\mathbf{w}}_M^{f\,t}(j) \;\; \overline{\mathbf{w}}_L^{b\,t}(j)]^t\,2^{\psi_j}, \quad (6)$$

where

$$\overline{\mathbf{w}}_M^f(j) = [\overline{w}_0^f(j)\;\overline{w}_1^f(j)\cdots\overline{w}_{M-1}^f(j)]^t \quad (7)$$

and

$$\overline{\mathbf{w}}_L^b(j) = [\overline{w}_1^b(j)\;\overline{w}_2^b(j)\cdots\overline{w}_L^b(j)]^t \quad (8)$$

are the mantissa vectors for the FFF and the FBF respectively for the j-th block. The integer ψ_j is a time-varying block exponent which needs to be updated at each block index j and is chosen to ensure that $|\overline{w}_m^f(j)| < \frac{1}{2}$ for $m \in Z_M$ and $|\overline{w}_{l+1}^b(j)| < \frac{1}{2}$ for $l \in Z_L$.

The Proposed Implementation: The proposed BFP treatment to the block ADFE consists of three stages that are mutually *pipelined*, namely,

(i) *Buffering* : Here, the input sequence $x(n)$ is partitioned into non-overlapping blocks of length N each, with the i-th block given by $\{x(n)|n \in Z_i'\}$, where $Z_i' = \{iN, iN+1, ..., iN+N-1\}$, $i \in Z$. For this, the input is shifted into a buffer of size N. We take N to be an integer multiple of Q, i.e., $N = KQ$, $K \in Z$, meaning that in each block of size N, the equalizer weights are updated K times. Also, we choose $N \geq M-1$, as otherwise, the input vector $\mathbf{x}(n)$ may involve data from three or more adjacent blocks and thus the complexity of implementation would go up. The buffer is cleared and its contents transferred to a block formatter once in every N input clock cycles.

(ii) *Block formatting of input*: Here, the data samples $x(n)$ constituting the i-th block, $i \in Z$ and available in FP form, are block formatted as per [3], resulting in the BFP representation : $x(n) = \overline{x}(n)\,2^{\gamma_i}$, $n \in Z_i'$ where $\gamma_i = ex_i + S_i$, $ex_i = \lfloor log_2 M_i \rfloor + 1$, $M_i = max\{|x(n)| \mid n \in Z_i'\}$. Next within the i-th block, consider the l-th sub-block, $l = 0, 1, \cdots, K-1$, given by the index set: $jQ+r$, $r = 0, 1, \cdots, Q-1$, where $j = iK + l$. For the l-th sub-block, the FFF output vector $\mathbf{y}_Q^f(jQ+Q-1)$ is then given as

$$\mathbf{y}_Q^f(jQ+Q-1) = X_{Q,M}\,\mathbf{w}_M^f(j), \quad (9)$$

$j = iK + l$. Assuming that each element of $X_{Q,M}$ belongs to the i-th block, we can express $\mathbf{y}_Q^f(jQ+Q-1)$ in a BFP form as

$$\mathbf{y}_Q^f(jQ+Q-1) = \overline{\mathbf{y}}_Q^f(jQ+Q-1)\,2^{\gamma_i+\psi_j}, \quad (10)$$

where

$$\overline{\mathbf{y}}_Q^f(jQ+Q-1) = \overline{X}_{Q,M}\,\overline{\mathbf{w}}_M^f(j) \quad (11)$$

denotes the FFF output mantissa vector for the l-th sub-block. For no overflow in $\overline{\mathbf{y}}_Q^f(jQ+Q-1)$, it is required that each element $|\overline{y}^f(jQ+r)| < 1, r \in Z_Q$. However, in the proposed scheme, we restrict each $\overline{y}^f(jQ+r), r \in Z_Q$ to lie between $+\frac{1}{4}$ and $-\frac{1}{4}$. From [3] and also from the fact that $|\overline{w}_m^f(n)| < \frac{1}{2}$, $m \in Z_M$, this implies a lower limit of S as $S_{min} = \lceil log_2 2M \rceil$. However, while S_{min} provides the lower limit of S_i, the actual value of S_i is, in fact, chosen to ensure a uniform BFP representation of $\mathbf{x}(n)$ during the block-to-block transition phase as well, i.e., when part of $\mathbf{x}(n)$ comes from the i-th block and part from the $(i-1)$-th block. This is realized by using the exponent assignment algorithm proposed in [2] and by rescaling the last $(M-1)$ elements of the previous block, namely, $x(iN-M+1), \cdots, x(iN-1)$, by dividing the respective mantissas by $2^{\Delta\gamma_i}$, where $\Delta\gamma_i = \gamma_i - \gamma_{i-1}$.

(iii) *Equalization and weight updating* : This consists of four main computations, namely,

(a) FFF output : The block formatter inputs $\overline{x}(n)$, $n \in Z_i'$, the rescaled mantissas for $x(iN-k)$, $k = 1, 2, \ldots, M-1$ and the block exponent γ_i to the FFF, which computes the output exponent for the l-th sub-block, $\gamma_i + \psi_j$, $j = iK + l$, $l = 0, 1, \cdots, K-1$, and the output mantissa vector as given by (10), using overlap-save method [6], FFT (denoted by 'F') and IFFT (denoted by 'F^{-1}'), as

$$\overline{\mathbf{y}}_Q^f(jQ+Q-1) = \mathbf{J}_Q\,[F^{-1}(\overline{X}_{S\times S}^d\,\overline{\mathbf{W}}_S^f)]_{last.Q} \quad (12)$$

where, $S = M+Q-1$,

$$\overline{\mathbf{W}}_S^f = F([\overline{\mathbf{w}}_M^{f\,t}(j) \;\; \mathbf{0}_{S-M}^t]^t) \quad (13)$$

and

$$\overline{X}^{d}_{S\times S} = diag(F([\overline{x}(jQ+Q-S)\cdots\overline{x}(jQ+Q-1)]^t)). \quad (14)$$

The matrix $\mathbf{J}_Q$ in (12) is the so-called exchange matrix, the FFT/IFFT in (12-14) itself is realized using BFP [7], where each butterfly computation is based on FxP operation only and up/down scaling is employed between the different butterfly stages to prevent overflow and also to use the dynamic range maximally.

(b) FBF output : Unlike the FFF, the computation of the FBF output vector $\mathbf{y}^b_Q(jQ+Q-1) = D_{Q,L}\,\mathbf{w}^b_L(j)$ contains unknown decisions in the matrix $D_{Q,L}$, as given by $d(k)$, $k = jQ,\cdots,jQ+Q-2$. To avoid this causality problem, we adopt the approach of [5], where the computation of $\mathbf{y}^b_Q(jQ+Q-1)$ is systematically decomposed into two parts: one containing past and thus known decisions, and the other involving purely the current and thus unknown decisions. For this, we first rewrite the FBF output $\mathbf{y}^b_Q(jQ+Q-1)$ as

$$\begin{aligned}\mathbf{y}^b_Q(jQ+Q-1)=&D^2_{Q,L-Q+1}\mathbf{w}^{b\,2}_{L-Q+1}(j)\\ &+W^b_{Q,2Q-1}(j)\mathbf{d}_{2Q-1}(jQ+Q-1)\end{aligned} \quad (15)$$

where, $W^b_{Q,2Q-1}(j)$ is a convolution matrix with the first row given as $[0\;\; w^b_1(j)\cdots w^b_{Q-1}(j)\;\; 0\cdots 0]$ and any m-th row, $m \geq 2$ given by m right shift of the first row. The vector $\mathbf{w}^{b\,2}_{L-Q+1}(j)$ is given by $\mathbf{w}^{b\,2}_{L-Q+1}(j) = [w^b_Q(j)\; w^b_{Q+1}(j)\cdots w^b_L(j)]^t$.

Partitioning $W^b_{Q,2Q-1}(j)$ as

$$W^b_{Q,2Q-1}(j) = [W^{b\,1}_{Q,Q}(j)\; W^{b\,2}_{Q,Q-1}(j)],$$

the FBF output can be written as

$$\begin{aligned}\mathbf{y}^b_Q(jQ+Q-1)=&D^2_{Q,L-Q+1}\mathbf{w}^{b\,2}_{L-Q+1}(j)\\ &+W^{b\,1}_{Q,Q}(j)\mathbf{d}_Q(jQ+Q-1)\\ &+W^{b\,2}_{Q,Q-1}(j)\mathbf{d}_{Q-1}(jQ-1),\end{aligned} \quad (16)$$

where $\mathbf{d}_Q(jQ+Q-1) = [d(jQ+Q-1)\cdots d(jQ)]^t$ contains the Q unknown decisions and $\mathbf{d}_{Q-1}(jQ-1) = [d(jQ-1)\cdots d(jQ-Q+1)]^t$ contains $Q-1$ known decisions from previous sub-blocks. Next, we group three terms on the R.H.S of (15) into two categories, namely, FB2 output given as,

$$\mathbf{y}^{b\,2}_Q(jQ+Q-1) = D^2_{Q,L-Q+1}\mathbf{w}^{b\,2}_{L-Q+1}(j), \quad (17)$$

and FB1 output given as,

$$\begin{aligned}\mathbf{y}^{b\,1}_Q(jQ+Q-1)=&W^b_{Q,2Q-1}(j)\mathbf{d}_{2Q-1}(jQ+Q-1) \quad (18)\\ =&\mathbf{y}^{b\,1,1}_Q(jQ+Q-1)+\mathbf{y}^{b\,1,2}_Q(jQ+Q-1)\end{aligned}$$

where

$$\mathbf{y}^{b\,1,1}_Q(jQ+Q-1) = W^{b\,1}_{Q,Q}(j)\,\mathbf{d}_Q(jQ+Q-1) \quad (19)$$

and

$$\mathbf{y}^{b\,1,2}_Q(jQ+Q-1) = W^{b\,2}_{Q,Q-1}(j)\,\mathbf{d}_{Q-1}(jQ-1) \quad (20)$$

Note that computation in (17) and (20) are simple convolutions involving known decisions and thus can be realized efficiently using overlap and save method/FFT. Now let

$$\begin{aligned}\mathbf{y}^c_Q(jQ+Q-1)=&\mathbf{y}^f_Q(jQ+Q-1)+\mathbf{y}^{b\,2}_Q(jQ+Q-1)\\ &+\mathbf{y}^{b\,1,2}_Q(jQ+Q-1)\end{aligned} \quad (21)$$

Then,

$$\mathbf{y}_Q(jQ+Q-1) = \mathbf{y}^c_Q(jQ+Q-1)+\mathbf{y}^{b\,1,1}_Q(jQ+Q-1). \quad (22)$$

Clearly, $\mathbf{y}^c_Q(jQ+Q-1)$ is the deterministic component in $\mathbf{y}_Q(jQ+Q-1)$, as it depends on available data and past decisions only, whereas, $\mathbf{y}^c_Q(jQ+Q-1)$ is the non-deterministic component which involve the unknown decisions.

In [5], an iterative procedure is suggested which first evaluates (19) (in time domain) by using an appropriately chosen initial value for $\mathbf{d}_Q(jQ+Q-1)$ and then, computes $\mathbf{y}_Q(jQ+Q-1)$ as per (22), which is then used in (2) to obtain the first iterate for $\mathbf{d}_Q(jQ+Q-1)$. This is again substituted in (17) and the iteration is carried out further. It is shown [5] that this iteration converges to the correct vector $\mathbf{d}_Q(jQ+Q-1)$ in Q or less number of steps for any choice of the initial value. A simple choice is to set the initial decision vector to the zero vector, i.e., we start the iterative procedure by replacing the unknown decisions in $D_{Q,L}$ with zeros.

Unlike the FFF, BFP computation of the FBF output $\mathbf{y}^b_Q(jQ+Q-1) = D_{Q,L}\,\mathbf{w}^b_L(j) \equiv D_{Q,L}\,\overline{\mathbf{w}}^b_L(j)2^{\psi_j}$ would require block formatting of the decision matrix $D_{Q,L}$ for each l-th sub-block, $l = 0,1,\cdots,K-1$, within the i-th block. However, like the FFF, the FBF output is also constrained to satisfy $|\overline{y}^b(jQ+r)| < \frac{1}{4}$, $r = 0,1,\cdots,Q-1$, $j = iK+l$, where $\overline{y}^b(jQ+r)$ denotes the mantissa of $y^b(jQ+r)$. The corresponding scaling factor, say, $S^{'}$ is then required to satisfy $S' \geq S^{'}_{min} = \lceil log_2 2L\rceil$. For minimum loss of bits, we choose $S' = S^{'}_{min}$. Now if the decisions $d(n), n \in Z$ generated by the quantizer are represented by $\beta(+1sign)$ bit FxP numbers, right shift by $S^{'}_{min}$ may, however, result in the loss of many significant bits, or, even flushing of the register to zero, for small values of $|d(n)|$. To avoid this, we first assume that the discrete levels of the quantizer are stored in a normalized, scaled format, meaning that the output of the quantizer, as per (2), is a length-Q decision vector in normalized FP form. The decision vector is subsequently block formatted with scaling factor $S = S^{'}_{min}$. Computation of $\mathbf{y}^b_Q(jQ+Q-1)$ in BFP then proceeds as follows:

• Assume that the past decisions present in the decision matrix $D_{Q,L}$, namely, $d(iQ-1),\cdots,d(jQ-L)$ are available in a BFP form as,

$$d(jQ-r) = \overline{d}_{j-1}(jQ-r)\,2^{\nu_{j-1}}, r = 1,2,\cdots,L, \quad (23)$$

with $|\overline{d}_{j-1}(jQ-r)| < 2^{-S^{'}_{min}}$. Then, (17) is evaluated as

$$\mathbf{y}^{b\,2}_Q(jQ+Q-1) = \overline{\mathbf{y}}^{b\,2}_Q(jQ+Q-1)2^{\nu_{j-1}+\psi_j}, \quad (24)$$

where, the mantissa vector $\overline{\mathbf{y}}^{b\,2}_Q(jQ+Q-1)$ is given as,

$$\overline{\mathbf{y}}^{b\,2}_Q(jQ+Q-1) = \overline{D}^2_{Q,L-Q+1}\overline{\mathbf{w}}^{b\,2}_{L-Q+1}(j), \quad (25)$$

with,

$$\overline{\mathbf{w}}_{L-Q+1}^{b\,2}(j) = [\overline{w}_Q^b(j)\ \overline{w}_{Q+1}^b(j)\cdots\overline{w}_L^b(j)]^t. \quad (26)$$

• Next, the term $\mathbf{y}_Q^{b\,1,2}(jQ+Q-1)$ in (20) is evaluated in BFP as

$$\mathbf{y}_Q^{b\,1,2}(jQ+Q-1) = \overline{\mathbf{y}}_Q^{b\,1,2}(jQ+Q-1)2^{\nu_{j-1}+\psi_j}, \quad (27)$$

where,

$$\overline{\mathbf{y}}_Q^{b\,1,2}(jQ+Q-1) = \overline{W}_{Q,Q-1}^{b\,2}\overline{\mathbf{d}}_{Q-1}(jQ-1) \quad (28)$$

.

• Using above, (21) is then computed in BFP as

$$\begin{aligned}\mathbf{y}_Q^c(jQ+Q-1)=&\overline{\mathbf{y}}_Q^f(jQ+Q-1)2^{\gamma_i+\psi_j}\\ &+[\overline{\mathbf{y}}_Q^{b\,2}(jQ+Q-1)\\ &+\overline{\mathbf{y}}_Q^{b\,1,2}(jQ+Q-1)]\,2^{\nu_{j-1}+\psi_j}\\ =&\overline{\mathbf{y}}_Q^c(jQ+Q-1)\,2^{\theta_j+\psi_j},\end{aligned} \quad (29)$$

where, the mantissa $\overline{\mathbf{y}}_Q^c(jQ+Q-1)$ and θ_j are evaluated as,

$if\ \gamma_i > \nu_{j-1}$

$$\begin{aligned}\overline{\mathbf{y}}_Q^c(jQ+Q-1)=&\overline{\mathbf{y}}_Q^f(jQ+Q-1)+[\overline{\mathbf{y}}_Q^{b\,2}(jQ+Q-1)\\ &+\overline{\mathbf{y}}_Q^{b\,1,2}(jQ+Q-1)]\,2^{\nu_{j-1}-\gamma_i},\\ \theta_j=&\gamma_i,\end{aligned}$$

$else$

$$\begin{aligned}\overline{\mathbf{y}}_Q^c(jQ+Q-1)=&\overline{\mathbf{y}}_Q^f(jQ+Q-1)2^{\gamma_i-\nu_{j-1}}\\ &+\overline{\mathbf{y}}_Q^{b\,2}(jQ+Q-1)+\overline{\mathbf{y}}_Q^{b\,1,2}(jQ+Q-1),\\ \theta_j=&\nu_{j-1}.\end{aligned}$$

It is easy to verify that $|\overline{y}^c(jQ+r)| < 1/2$.

• As explained above, the solution for correct $\mathbf{d}_Q(jQ+Q-1)$ is obtained iteratively. Assume that at each k-th step of iteration, $0 \le k \le Q$, the k-th iterate for $\mathbf{d}_Q(jQ+Q-1)$, namely, $\mathbf{d}_Q^k(jQ+Q-1)$ is available in a BFP form as $\mathbf{d}_Q^k(jQ+Q-1) = \overline{\mathbf{d}}_Q^k(jQ+Q-1)2^{\tau_{k,j}}$ with $|\overline{\mathbf{d}}_Q^k(jQ+r)| < 2^{-S'_{min}}$, $r \in Z_Q$. The iteration steps are then as follows:

The initial condition for the iteration is set as: $\overline{\mathbf{d}}_Q^0(jQ+Q-1) = \mathbf{0}_Q$, $\tau_{0,j}$: a large negative number. Then, for any k, $0 \le k < Q$, evaluate (19) as

$$\begin{aligned}\mathbf{y}_Q^{b\,1,1,k}(jQ+Q-1)=&\overline{W}_{Q,Q}^{b\,1}(j)\,\overline{\mathbf{d}}_Q^k(jQ+Q-1)\,2^{\tau_{k,j}+\psi_j}\\ \equiv&\overline{\mathbf{y}}_Q^{b\,1,1,k}(jQ+Q-1)\,2^{\tau_{k,j}+\psi_j}.\end{aligned}$$

• The k-th iterate of the pre-decision output $\mathbf{y}_Q^k(jQ+Q-1)$, as per (22) is then obtained as

$$\begin{aligned}\mathbf{y}_Q^k(jQ+Q-1)=&\overline{\mathbf{y}}_Q^c(jQ+Q-1)\,2^{\theta_j+\psi_j}\\ &+\overline{\mathbf{y}}_Q^{b\,1,1,k}(jQ+Q-1)\,2^{\tau_{k,j}+\psi_j}\\ =&\overline{\mathbf{y}}_Q^k(jQ+Q-1)\,2^{\xi_j+\psi_j}.\end{aligned} \quad (30)$$

Where,

$if\ \theta_j > \tau_{k,j}$

$$\begin{aligned}\overline{\mathbf{y}}_Q^k(jQ+Q-1)=&\overline{\mathbf{y}}_Q^c(jQ+Q-1)\\ &+\overline{\mathbf{y}}_Q^{b\,1,1,k}(jQ+Q-1)2^{\tau_{k,j}-\theta_j},\ \ \xi_j=\theta_j,\end{aligned}$$

$else$

$$\begin{aligned}\overline{\mathbf{y}}_Q^k(jQ+Q-1)=&\overline{\mathbf{y}}_Q^c(jQ+Q-1)2^{\theta_j-\tau_{k,j}}\\ &+\overline{\mathbf{y}}_Q^{b\,1,1,k}(jQ+Q-1),\ \ \xi_j=\tau_{k,j}.\end{aligned}$$

It is easy to verify that $|\overline{y}^k(jQ+r)| < \frac{1}{2}$, $r \in Z_Q$, where, $\overline{y}^k(jQ+r) = [\mathbf{J}_Q\,\overline{\mathbf{y}}_Q^k(jQ+Q-1)]_r$. Substituting $\mathbf{y}_Q^k(jQ+Q-1)$ in (2), the $(k+1)$-th iterate $\mathbf{d}_Q^{k+1}(jQ+Q-1)$ is then obtained, which is subsequently block formatted in order to be represented as $\mathbf{d}_Q^{k+1}(jQ+Q-1) = \overline{\mathbf{d}}_Q^{k+1}(jQ+Q-1)\,2^{\tau_{k+1,j}}$.

• If for any k, $0 \le k < Q$, $\mathbf{d}_Q^{k+1}(jQ+Q-1) \equiv \mathbf{d}_Q^k(jQ+Q-1)$, then convergence is reached, meaning that $\mathbf{d}_Q(jQ+Q-1) = \mathbf{d}_Q^k(jQ+Q-1)$, i.e., the iteration can be terminated and $\mathbf{y}_Q(jQ+Q-1)$ in (3) can be taken as $\mathbf{y}_Q^k(jQ+Q-1)$. The iteration is guaranteed to converge in Q or less number of steps [5].

• Once the iteration converges , say, in k steps, the two vectors, $\mathbf{d}_Q(jQ+Q-1) = \mathbf{d}_Q^k(jQ+Q-1)2^{\tau_{k,j}}$ and $\mathbf{d}_L(jQ-1) = [\overline{d}_{j-1}(jQ-1),\cdots,\overline{d}_{j-1}(jQ-L)]2^{\nu_{j-1}}$ are to be jointly block formatted to produce $[\overline{d}_j(jQ+Q-1),\cdots,\overline{d}_j(jQ-1),\cdots,\overline{d}_j(jQ-L)]\,2^{\nu_j}$, for use during weight updating and also during processing of the $(l+1)$-th sub-block.

(c) Error : The error vector $\mathbf{e}_Q(jQ+Q-1)$ as given by (3), is computed as

$$\begin{aligned}\mathbf{e}_Q(jQ+Q-1)=&\mathbf{d}_Q(jQ+Q-1)-\mathbf{y}_Q(jQ+Q-1)\\ =&\overline{\mathbf{d}}_Q(jQ+Q-1)2^{\nu_j}\\ &-\overline{\mathbf{y}}_Q(jQ+Q-1)2^{\xi_j+\psi_j},\end{aligned}$$

where,

$$\xi_j = max\{\gamma_i, \nu_{j-1}, \tau_{k,j}\}, \quad (31)$$

with k denoting the iteration index at convergence. Note that on convergence, $\tau_{k+1,j} \equiv \tau_{k,j}$ and thus, $\nu_j = max\{\nu_{j-1}, \tau_{k+1,j}\} = max\{\nu_{j-1}, \tau_{k,j}\}$, meaning $\xi_j \ge \nu_j$. Thus

$$\begin{aligned}\mathbf{e}_Q(jQ+Q-1)=&[\overline{\mathbf{d}}_Q(jQ+Q-1)2^{\nu_j-\xi_j-\psi_j}\\ &-\overline{\mathbf{y}}_Q(jQ+Q-1)]2^{\xi_j+\psi_j}\\ =&\overline{\mathbf{e}}_Q(jQ+Q-1)2^{\xi_j+\psi_j},\end{aligned}$$

where,

$$\overline{\mathbf{e}}_Q(jQ+Q-1) = \overline{\mathbf{d}}_Q(jQ+Q-1)2^{\nu_j-\xi_j-\psi_j} - \overline{\mathbf{y}}_Q(jQ+Q-1) \quad (32)$$

Using the fact that $|\overline{y}(jQ+r)| < 1/2$, $r \in Z_Q$, it is easy to verify that $|\overline{e}(jQ+r)| < 1$.

(d)Weight updating : For updating $\mathbf{w}(j)$, we first try to express $\mathbf{w}_M^f(j+1)$ and $\mathbf{w}_L^b(j+1)$ as $\mathbf{w}_M^f(j+1) = \overline{\mathbf{u}}_M^f(j)2^{\psi_j}$ and $\mathbf{w}_L^b(j+1) = \overline{\mathbf{u}}_L^b(j)2^{\psi_j}$ for some appropriate $\overline{\mathbf{u}}_M^f(j) = [\overline{u}_0^f(j),\cdots,\overline{u}_{M-1}^f(j)]^t$ and $\overline{\mathbf{u}}_L^b(j) = [\overline{u}_1^b(j),\cdots,\overline{u}_L^b(j)]^t$ that are constrained as $|\overline{u}_m^f(j)| < 1$, $|\overline{u}_{l+1}^b(j)| < 1$, $m \in$

Z_M, $l \in Z_L$. Then, if each $\overline{u}^f_m(j)$ and each $\overline{u}^b_{l+1}(j)$ lie within $\pm\frac{1}{2}$, we make the assignments:

$$\overline{\mathbf{w}}(j+1) = [\overline{\mathbf{u}}^f_M(j) \;\; \overline{\mathbf{u}}^{\mathbf{b}}_{\mathbf{L}}(j)]^t, \quad \psi_{j+1} = \psi_j. \tag{33}$$

Otherwise, we scale down $\overline{\mathbf{u}}^f_M(j)$ and $\overline{\mathbf{u}}^b_L(j)$ by 2, meaning,

$$\overline{\mathbf{w}}(j+1) = \frac{1}{2}[\overline{\mathbf{u}}^f_M(j) \;\; \overline{\mathbf{u}}^b_L(j)]^t, \quad \psi_{j+1} = \psi_j + 1. \tag{34}$$

Substituting $X_{Q,M}$, $D_{Q,L}$ and $\mathbf{e}_Q(jQ+Q-1)$ in (4) and (5) by $\overline{X}_{Q,M}2^{\gamma_i}$, $\overline{D}_{Q,L}2^{\nu_j}$ and $\overline{\mathbf{e}}_Q(jQ+Q-1)2^{\xi_j+\psi_j}$ respectively, we can write,

$$\overline{\mathbf{u}}^f_M(j) = \overline{\mathbf{w}}^f_M(j) \; + \mu\Sigma_{r=0}^{Q-1}\overline{\mathbf{x}}_M(jQ+r)\,\overline{e}(jQ+r)2^{\gamma_i+\xi_j}, \tag{35}$$

and

$$\overline{\mathbf{u}}^b_L(j) = \overline{\mathbf{w}}^b_L(j) + \mu\Sigma_{r=0}^{Q-1}\overline{\mathbf{d}}_L(jQ+r-1)\overline{e}(jQ+r)2^{\nu_j+\xi_j}. \tag{36}$$

Since $|\overline{w}^f_m| < 1/2$, $m \in Z_M$ and $|\overline{w}^b_{l+1}| < 1/2$, $l \in Z_L$, it is enough to have
(A) $\mu\Sigma_{r=0}^{Q-1}|\overline{x}_M(jQ+r-m)|\,|\overline{e}(jQ+r)|2^{\gamma_i+\xi_j} \le \frac{1}{2}$,
(B) $\mu\Sigma_{r=0}^{Q-1}|\overline{d}_L(jQ+r-l-1)|\,|\overline{e}(jQ+r)|\,2^{\nu_j+\xi_j} \le \frac{1}{2}$,
in order to satisfy $|\overline{u}^f_m(j)| < 1$, $m \in Z_M$ and $|\overline{u}^b_{l+1}(j)| < 1$, $l \in Z_L$ respectively. Now, for $r \in Z_Q$,

$$\begin{aligned}|\overline{e}(jQ+r)|\,2^{\xi_j+\psi_j} \equiv & |e(jQ+r)| \\ \le & |\overline{d}(jQ+r)|\,2^{\nu_j} + |\overline{y}(jQ+r)|\,2^{\xi_j+\psi_j} \\ \le & 2^{\nu_j - S'_{min}} + |\overline{y}(jQ+r)|2^{\xi_j+\psi_j}.\end{aligned}$$

Again,

$$\begin{aligned}|\overline{y}(jQ+r)|\,2^{\xi_j+\psi_j} \equiv & |y(jQ+r)| \\ \le & |\overline{y}^f(jQ+r)|\,2^{\gamma_i+\psi_j} \\ & + |\overline{y}^b(jQ+r)|\,2^{\nu_j+\psi_j} \\ \le & \frac{M}{2}\,2^{\gamma_i+\psi_j-S_i} + \frac{L}{2}\,2^{\nu_j+\psi_j-S'_{min}},\end{aligned}$$

meaning

$$|\overline{e}(jQ+r)|\,2^{\xi_j+\psi_j} \le 2^{\nu_j-S'_{min}}[1 + \frac{M}{2}\,2^{\gamma_i+\psi_j-S_i} + \frac{L}{2}\,2^{\psi_j}]. \tag{37}$$

Substituting (37) in (A) and noting that $|\overline{x}_M(jQ+r-m)|\,2^{\gamma_i} < 2^{ex_i}$, it is then sufficient to have,

$$\mu \le \frac{1}{Q[2^{ex_i+\nu_j-\psi_j-S'_{min}+1} + M2^{2ex_i} + L2^{ex_i+\nu_j-S'_{min}}]} \tag{38}$$

From above, we obtain a general upper bound for μ by equating ν_j to its highest value of $\beta+1+S'_{min}$, ψ_j to its minimum value of zero, and replacing ex_i by $ex_{max} = max\{ex_i | i \in Z\}$. The general upper bound is given by:

$$\mu \le \mu^f = \frac{1}{Q[2^{ex_{max}+\beta+2} + M2^{2ex_{max}} + L2^{ex_{max}+\beta+1}]} \tag{39}$$

Similarly, from (37), condition (B) and recalling that $\overline{d}_L(jQ+r-l-1)| < 2^{-S'_{min}}, l \in Z_L$, the general upper bound for FBF can be obtained as

$$\mu \le \mu^b = \frac{1}{Q[2^{2\beta+3} + M\,2^{ex_{max}+\beta+1} + L\,2^{2\beta+2}]} \tag{40}$$

The final choice of μ will be made following $\mu \le min\{\mu^f, \mu^b\}$. The two bounds, μ^f and μ^b are easily seen to be related by a simple constant, i.e., $\frac{\mu^f}{\mu^b} = 2^{ex_{max}-\beta-1}$.

III. Complexity Issues

The proposed schemes rely mostly on FxP arithmetic, resulting in computational complexities much less than that of their FP-based counterparts. In the following, we provide a comparative account of both the approaches in terms of complexity. Consider the computation of the FFF output mantissa at any n-th ($n = jQ+r, r \in Z_Q$ index, as given by (9). Clearly, in the proposed treatment, this requires M "Multiply and Accumulate (MAC)" operations (FxP) and at the most, one exponent addition operation to compute the exponent $\gamma_i + \psi_j$. In contrast, in a FP-based realization, this would require M FP-based MAC operations. Table I provides a comparative account of the two approaches in terms of number of operations required per iteration. It is easy to verify from Table I that given a low cost, simple FxP processor with single cycle MAC and barrel shifter units, the proposed scheme is about *four times faster* than a FP based implementation, for moderately large values of M, L and Q.

TABLE I

A comparison between the BFP vis-à-vis the FP-based realizations of the Block ADFE. Number of operations required per iteration for (a) weight updating, and (b) filtering are shown. [$R = L + M$, $T = R + 2Q(I+1)$, MAC : Multiply and Accumulate, MC : Magnitude Check, EC : Exponent Comparison, EA : Exponent Addition.]

(a)	MAC	Shift	MC	EC	EA
BFP	$(R+2)Q$	$P+3Q+1$	R	Nil	2
FP	$(R+1)Q$	$2(R+1)Q$	Nil	$(R+1)Q$	$(R+1)Q$

(b)	MAC	Shift	EC	EA	Add
BFP	$R+QI$	Q(2I+3)+L-I-1	QI+2	$I+3$	$Q(3+I)$
FP	T	$2(T+Q)$	$(T+Q)$	$2T+Q)$	Q

References

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

[2] A. Mitra, M. Chakraborty and H. Sakai, "A Block Floating Point treatment to the LMS Algorithm: Efficient realization and roundoff error analysis", *IEEE Trans. Signal Processing*, pp. 4536-4544, Dec. 2005.

[3] M.Chakraborty, R. Shaik and Moon Ho Lee, "A Block Floating Point Realization of the Block LMS Algorithm", *IEEE Trans. on Circuits Syst., part II*, Vol. 53, no. 9, pp. 812-816, September 2006.

[4] R. Shaik and M. Chakraborty, "An Efficient Finite Precision Realization of the Adaptive Decision Feedback Equalizer", in Proc. *2007 IEEE International Symposium on Circuits And Systems (ISCAS)*, New Orleans, USA, May. 2007, pp. 1341-1344.

[5] K Berberidis and P Karaivazoglou, "An efficient block adaptive decision feedback equalizer implemented in the frequency domain," *IEEE trans. Signal Processing*, vol. 50, No. 9, pp. 2273-2285, Sept. 2002.

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

[7] D. Elam and C. Lovescu, "A Block Floating Point Implementation for an N-Point FFT on the TMS320C55X DSP", *Texas Instruments Application Report*, SPRA 948, Sept., 2003.

[8] S. Haykin, *Adaptive Filter Theory*, Englewood Cliffs, NJ: Prentice-Hall, 1986.