Signal Processing 93 (2013) 1162–1171

Contents lists available at SciVerse ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

A block floating point treatment to finite precision realization of the adaptive decision feedback equalizer

Rafi Ahamed Shaik [a], Mrityunjoy Chakraborty [b,*]

[a] Department of Electronics and Electrical Engineering,, Indian Institute of Technology, Guwahati-781 039, India
[b] Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology, Kharagpur-721 302, India

ARTICLE INFO

Article history:
Received 6 July 2012
Received in revised form
23 November 2012
Accepted 26 November 2012
Available online 4 December 2012

Keywords:
Block floating point
Adaptive decision feedback equalizer
Mantissa
Exponent
Scaling
Overflow

ABSTRACT

A scheme for efficient realization of the adaptive decision feedback equalizer (ADFE) is presented using the block floating point (BFP) data format which enables the ADFE to process rapidly varying data over a wide dynamic range, at a fixed point like complexity. The proposed scheme adopts appropriate BFP format for the data as well as the filter weights and works out separate update relations for the filter weight mantissas and exponents. Overflows at the feed forward and the feedback filter output are prevented by certain dynamic scaling of the respective input, while overflow in weight update calculations is avoided by imposing certain upper bound on the algorithm step size μ which is shown to be less than the convergence bound. The proposed scheme deploys mostly simple fixed point operations and is shown to achieve considerable computational gain over its floating point based counterpart.

1. Introduction

In many communication systems, one often encounters channels with long impulse response (IR) that results in inter symbol interference (ISI) over several symbol periods. A typical example is given by multipath channels, where under the same delay spread conditions, the channel IR length increases with increase in symbol transmission rate. Also, on many occasions, one comes across channels that exhibit spectral nulls. The linear equalizer is not a very effective option in such cases for cancelation of the ISI, due to very large order requirement and also due to the possibility of substantial noise enhancement by the spectral peaks of the equalizer. A more effective solution in such cases is provided by the adaptive decision feedback equalizer (ADFE). 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, e.g., the LMS algorithm [1]. In practice, however, the ADFE is often required to operate in a resource constrained (e.g., low power, low chip area) environment while maintaining high throughput rate. This makes it important to devise methods for reducing the complexity of the ADFE realization. Several attempts such as [2–5] have come up in recent years which try to meet this objective by means of suitable algorithmic and architectural transformations. In this paper, we propose a different approach to the complexity reduction by adopting suitable data format for the input and the equalizer coefficients.

In a practical communication receiver, the received signal level is usually very weak which also fluctuates randomly due to effects like fading. In such cases, the input to the ADFE is obtained by first processing the received sample through a

* Corresponding author. Tel.: +91 9434026874; fax: +91 3222 255303.
E-mail addresses: rafiahamed@iitg.ernet.in (R.A. Shaik), mrityun@ece.iitkgp.ernet.in (M. Chakraborty).

0165-1684/$ - see front matter
http://dx.doi.org/10.1016/j.sigpro.2012.11.024

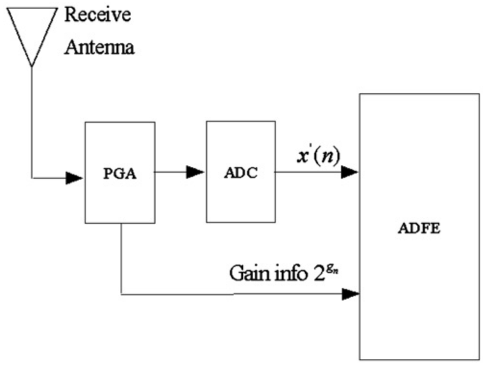

Fig. 1. Front end of an adaptive decision feedback equalizer.

programmable gain amplifier (PGA) as shown in Fig. 1. The PGA continuously adjusts its gain (by a power of two) so as to utilize the available dynamic range of the ADC maximally. This, however, gives rise to a floating point (FP) representation of the ADFE input, with the mantissa and the exponent given respectively by the ADC output and the negative of the PGA gain. A direct FP based implementation of the ADFE would, however, require much higher processing complexity than a typical fixed point (FxP) based realization, as, in FP, each stage of computation requires several additional steps not required in FxP. The block floating point (BFP) data format, in this context, is a viable alternative to the FP system. 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 provides a FP like high dynamic range. Over decades, the BFP format has been used for efficient implementation of many signal processing algorithms. These include various forms of fixed coefficient digital filters [6–11], adaptive filters [12] and unitary transforms [13–15] on one hand and several audio data transmission standards like NICAM (stereophonic sound system for PAL TV standard), the audio part of MUSE (Japanese HDTV standard) and DSR (German Digital Satellite Radio System) on the other.

In this paper, we present a BFP treatment to finite precision implementation of the LMS based ADFE[1]. Such a realization is intrinsically more difficult than a BFP based realization [12] of transversal adaptive filters, since, unlike the latter, the ADFE consists of a decision feedback loop with a non-linear decision device. The proposed scheme provides a viable solution to this problem by effectively modifying and extending the framework of [12]. For this, first, appropriate BFP formats are adopted for the FFF and the FBF coefficients. Separate update relations for the mantissas as well as the exponents for each set of coefficients are worked out next. For the FFF, the input is block formatted by an efficient block formatting algorithm which also includes certain dynamic scaling of the data for preventing overflow at the FFF output. For the FBF, however, no block processing of the corresponding input (i.e., decisions) is possible, as that makes the system non-causal. Instead, the data stored in the FBF memory is block formatted at each time index, by appropriately modifying the proposed block formatting algorithm. It is also required to prevent overflow in the weight update computations of the FFF and the FBF. This gives rise to two upper bounds for the step size μ, one coming from the FFF and the other from the FBF considerations. The two bounds are related by a simple constant and the lesser of them is used as an upper limit of μ, which is interestingly seen to be less than $2/\mathrm{tr}\,\mathbf{R}$, (where $\mathbf{R}$ is the autocorrelation matrix of the input signal $x(n)$ and is given by $R=E[\mathbf{x}(\mathbf{n})\mathbf{x}^t(\mathbf{n})]$), i.e., upper bound of μ for convergence of the LMS iteration. The proposed scheme relies largely on simple FxP operations and thus achieves considerable speed up over a direct FP based realization. Also, simulation results show no appreciable degrading effect on the ADFE performance due to block formatting of data and filter coefficients in finite precision.

The organization of the paper is as follows: Section 2 presents a background of the BFP concept. Section 3 presents the proposed implementation scheme where BFP treatments to all the computational stages are worked out in detail. Computational complexity analysis of all the schemes proposed is carried out in Section 4 showing superiority of the proposed method over traditional FP based implementation. Simulation studies on the effects of block formatting on ADFE performance are presented in Section 5. Throughout the paper, characters with an overbar are used to indicate mantissas and the symbol Z_m for any integer m, $m \geq 0$ is used to denote the set $\{0,1,\ldots,m-1\}$.

2. BFP background

The BFP representation can be viewed as a special case of the FP format, where every f-overlapping block of N incoming data has a joint scaling factor determined by the data sample with the highest magnitude in the block. In other words, given a block $[x_0,\ldots,x_{N-1}]$, we represent it in BFP as $[x_0,\ldots,x_{N-1}]=[\overline{x}_0,\ldots,\overline{x}_{N-1}]2^{\gamma}$ where $\overline{x}_l(=x_l2^{-\gamma})$ represents the mantissa of x_l for $l\in Z_N$ and the block exponent γ is defined as $\gamma=\lfloor\log_2 Max\rfloor+1+S$ where $Max=max(|x_0|,\ldots,|x_{N-1}|)$,

[1] Some preliminary results of this paper had been presented by the authors at ISCAS-2007, New Orleans, USA, 2007 [16].