

A Pipelined Strength-Reduced Adaptive Filter : Finite Precision Analysis and Application to 155.52 Mb/s ATM-LAN

Manish Goel and Naresh R. Shanbhag
Coordinated Science Laboratory/ECE Department,
Univ. of Illinois at Urbana-Champaign
1308 West Main Street, Urbana, IL-61801.

Abstract—In this paper, we present the finite-precision analysis of the pipelined strength-reduced adaptive filter architecture. This architecture provides the dual advantage low power dissipation and high speed operation. Precision requirements for the traditional cross-coupled (CC) and the strength-reduced (SR) architectures are compared. In case of the filter block (F-block) coefficient precision, the SR architecture requires 0.3 bits more than the corresponding block in the CC architecture. Similarly, the weight-update (WUD-block) in the SR architecture is shown to require 0.5 bits fewer than the corresponding block in the CC architecture. This finite-precision architecture is then used as a near-end crosstalk (NEXT) canceller for 155.52 Mb/s ATM-LAN over unshielded twisted pair (UTP) category-3 cable. Simulation results are presented in support of the analysis.

I. INTRODUCTION

Strength reduction is an algebraic transformation, which has been proposed [3] to trade-off multipliers with adders in a complex multiplication thereby achieving power reduction. In [6], we proposed the application of *strength reduction* transformation at the algorithmic level to adaptive systems involving complex signals and filters. It was shown in [6] that the strength-reduced (SR) filter enables power savings of 21 – 25% over the traditional cross-coupled (CC) with no loss in performance. However, the application of strength reduction increases the critical path and hence an inherently pipelined SR (PIPSR) architecture was also presented. Furthermore, by trading of the throughput gained through pipelining with power supply scaling [3], it was demonstrated that additional power savings of 40 – 69% are feasible. In this paper, we present the finite-precision analysis of the PIPSR architecture developed in [6]. It is shown that the precision requirements of SR architecture are similar to that of the CC architecture. Clearly, the SR and PIPSR architectures are attractive alternatives to the traditional CC architecture for high bit-rate communications and digital signal processing applications.

In this paper, a linear model is employed for coefficient

quantization noise. The filter (F) block precision, B_F is chosen such that the signal-to-quantization-noise-ratio (SQNR) is greater than the desired signal-to-noise ratio, SNR_o . The coefficient precision for weight-update (WUD) block, B_{WUD} is determined by applying the *stopping criterion* [2], [4], which puts a lower limit upon the correction term being added to the weight update. This criterion is given by

$$\mu^2 E[|e(n)|^2] \sigma^2 \geq 2^{-2B_{WUD}}, \quad (1.1)$$

where μ is the step-size, $E[|e(n)|^2]$ the mean-squared error, σ^2 is the power of the received signal and B_{WUD} is the precision (including sign-bit) of the coefficients in weight-update block (WUD-block). A *non-linear* analysis is presented in [1] for a tighter bound on B_{WUD} . Such a model however becomes complex to employ if the number of terms in the weight-update equation increases as is the case with CC and SR architectures. The purpose of this paper is just to compare the precision requirements for CC and SR architectures. We employ linear-analysis for the comparison.

We demonstrate an application of the finite-precision SR architecture as a near-end crosstalk (NEXT) canceller for 155.52 Mb/s [5] ATM-LAN over 100 meters of unshielded twisted pair category-3 (UTP-3) cable employing 64-CAP (carrierless amplitude/phase) modulation scheme. We present the simulation results for this application in order to determine the precision requirements of various signals and to support the analytical results presented in the paper.

II. PIPELINED STRENGTH-REDUCED (PIPSR) ARCHITECTURE

In this section, we review the strength reduction transformation and development of the PIPSR architecture [6] from the CC architecture. The reader is referred to [6] for more details, while we will present only the final results here.

A. Strength Reduction Transformation

Consider the problem of computing the product of two complex numbers $(a + jb)$ and $(c + jd)$ as shown below

$$(a + jb)(c + jd) = (ac - bd) + j(ad + bc). \quad (2.1)$$

This research was supported by Analog Devices, Inc. and National Science Foundation CAREER award MIP-9623737.

From (2.1), a direct-mapped architectural implementation would require a total of four real multiplications and two real additions to compute the complex product. Application of strength reduction involves reformulating (2.1) as follows

$$(a-b)d+a(c-d) = ac-bd, \quad (a-b)d+b(c+d) = ad+bc, \quad (2.2)$$

where we see that strength reduction reduces the number of multipliers by one at the expense of three additional adders. Typically, multiplications are more expensive than additions and hence we achieve an overall savings in hardware.

B. Strength-reduced (SR) Architecture

The **SR** architecture [6] is obtained by applying strength reduction transformation at the algorithmic level instead of at the multiply-add level described in the previous subsection. Starting with the complex LMS [8] algorithm, assume that the filter input is a complex signal vector $\mathbf{X}(n)$ given by $\mathbf{X}(n) = \mathbf{X}_r(n) + j\mathbf{X}_i(n)$, where $\mathbf{X}_r(n)$ and $\mathbf{X}_i(n)$ are the real and the imaginary parts. Furthermore, if the filter $\mathbf{W}(n)$ is also complex ($\mathbf{W}(n) = \mathbf{c}(n) + j\mathbf{d}(n)$), then the complex LMS algorithm is given by

$$y(n) = \mathbf{W}^H(n-1)\mathbf{X}(n), \quad \mathbf{W}(n) = \mathbf{W}(n-1) + \mu e^*(n)\mathbf{X}(n), \quad (2.3)$$

where μ is the step-size, $e(n) = d(n) - y(n)$ is the error, $d(n)$ is the desired signal and $\mathbf{W}(n)$ is the coefficient vector. Also, $e^*(n)$ represents the complex conjugate of the signal $e(n)$ and $\mathbf{W}^H(n)$ represents the hermitian (complex conjugate transpose) of $\mathbf{W}(n)$.

From (2.3), we see that there are two complex multiplications/inner-products involved. Traditionally, the complex LMS algorithm is implemented via the **CC** architecture, which is described by the following equations:

$$y_r(n) = \mathbf{c}^T(n-1)\mathbf{X}_r(n) + \mathbf{d}^T(n-1)\mathbf{X}_i(n) \quad (2.4a)$$

$$y_i(n) = \mathbf{c}^T(n-1)\mathbf{X}_i(n) - \mathbf{d}^T(n-1)\mathbf{X}_r(n) \quad (2.4b)$$

$$\mathbf{c}(n) = \mathbf{c}(n-1) + \mu [e_r(n)\mathbf{X}_r(n) + e_i(n)\mathbf{X}_i(n)] \quad (2.4c)$$

$$\mathbf{d}(n) = \mathbf{d}(n-1) + \mu [e_r(n)\mathbf{X}_i(n) - e_i(n)\mathbf{X}_r(n)], \quad (2.4d)$$

where $e(n) = e_r(n) + je_i(n)$ and the **F**-block output is given by $y(n) = y_r(n) + jy_i(n)$. Equations (2.4a-2.4b) and (2.4c-2.4d) define the computations in the **F**-block and the **WUD**-block, respectively. A direct-mapped implementation of (2.4) would require $8N$ multipliers and adders.

We see that (2.4) has two complex inner-products and hence can benefit from the application of strength reduction. Doing so results in the following equations, which describe the **F**-block computations of the **SR** architecture [6]:

$$y_1(n) = \mathbf{c}_1^T(n-1)\mathbf{X}_r(n), \quad y_2(n) = \mathbf{d}_1^T(n-1)\mathbf{X}_i(n),$$

$$y_3(n) = -\mathbf{d}^T(n-1)\mathbf{X}_1(n), \quad (2.5a)$$

$$y_r(n) = y_1(n) + y_3(n), \quad y_i(n) = y_2(n) + y_3(n), \quad (2.5b)$$

where $\mathbf{X}_1(n) = \mathbf{X}_r(n) - \mathbf{X}_i(n)$, $\mathbf{c}_1(n) = \mathbf{c}(n) + \mathbf{d}(n)$, and $\mathbf{d}_1(n) = \mathbf{c}(n) - \mathbf{d}(n)$. Similarly, the **WUD** computation is described by,

$$\mathbf{c}_1(n) = \mathbf{c}_1(n-1) + \mu[\mathbf{e}\mathbf{X}_1(n) + \mathbf{e}\mathbf{X}_3(n)] \quad (2.6a)$$

$$\mathbf{d}_1(n) = \mathbf{d}_1(n-1) + \mu[\mathbf{e}\mathbf{X}_2(n) + \mathbf{e}\mathbf{X}_3(n)], \quad (2.6b)$$

where $\mathbf{e}\mathbf{X}_1(n) = 2e_r(n)\mathbf{X}_i(n)$, $\mathbf{e}\mathbf{X}_2(n) = 2e_i(n)\mathbf{X}_r(n)$, $\mathbf{e}\mathbf{X}_3(n) = e_1(n)\mathbf{X}_1(n)$, $e_1(n) = e_r(n) - e_i(n)$. It is easy to show that the **SR** architecture requires only $6N$ multipliers and $8N + 3$ adders. This is the reason why the **SR** architecture results in 21–25% power savings [6] over the **CC** architecture.

C. Pipelined Strength-reduced (PIPSR) Architecture

As explained in [6], both the **SR** as well as **CC** architectures are bounded by a maximum possible clock rate due the computations in this critical path. This throughput limitation is eliminated via the application of the *relaxed look-ahead transformation* [7] to the **SR** architecture (see (2.5-2.6)). Application of relaxed look-ahead to the **SR** architecture in (2.5-2.6) results in the following equations that describe the **F**-block computations in the **PIPSR** architecture:

$$y_1(n) = \mathbf{c}_1^T(n - D_2)\mathbf{X}_r(n), \quad y_2(n) = \mathbf{d}_1^T(n - D_2)\mathbf{X}_i(n),$$

$$y_3(n) = -\mathbf{d}^T(n - D_2)\mathbf{X}_1(n), \quad (2.7a)$$

$$y_r(n) = y_1(n) + y_3(n), \quad y_i(n) = y_2(n) + y_3(n), \quad (2.7b)$$

where D_2 is the number of delays introduced before feeding the filter coefficients into the **F**-block. Similarly, the computation of the **WUD** block of the **PIPSR** architecture are given by

$$\mathbf{c}_1(n) = \mathbf{c}_1(n - D_2) + \mu \sum_{i=0}^{LA-1} [\mathbf{e}\mathbf{X}_1(n - D_1 - i) + \mathbf{e}\mathbf{X}_3(n - D_1 - i)] \quad (2.8a)$$

$$\mathbf{d}_1(n) = \mathbf{d}_1(n - D_2) + \mu \sum_{i=0}^{LA-1} [\mathbf{e}\mathbf{X}_2(n - D_1 - i) + \mathbf{e}\mathbf{X}_3(n - D_1 - i)], \quad (2.8b)$$

where $\mathbf{e}\mathbf{X}_1(n)$, $\mathbf{e}\mathbf{X}_2(n)$ and $\mathbf{e}\mathbf{X}_3(n)$ are defined in the previous subsection, $D_1 \geq 0$ are the delays introduced into the error feedback loop and $0 < LA \leq D_2$ indicates the number of terms considered in the sum-relaxation. A block level implementation of the **PIPSR** architecture is shown in Fig. 1 (see [6] for details) where D_1 and D_2 delays will be employed to pipeline the various operators such as adders and multipliers at a fine-grain level. The high-throughput of the **PIPSR** architecture can be traded-off with supply voltage reduction resulting in additional power savings [6] of 40–69%. Therefore, the **PIPSR** architecture results in 60–90% power savings as compared to the serial **CC** architecture.

III. FINITE PRECISION ANALYSIS

In this section, we will present a comparison of the precision requirements of the **CC** and **SR** architectures. We employ linear models [2] for the quantization noise. Further, **F**-block coefficient precision, B_F is determined by treating **F**-block as a constant coefficient FIR filter and choosing $SQNR \gg SNR$. The stopping criterion [2] is used for determining the **WUD**-block coefficient precision, B_{WUD} .

A. F-block Precision

Define $B_{x,y}$ to be the coefficient precision (including sign-bit) in **x** block of **y** architecture. Also, let N be the tap-length, J_{float} be the floating point *MSE* and J_o to be the specified *MSE* for the fixed-point algorithm.

Now, we determine the quantization error due to finite-precision implementation of the **F**-block. For **CC** architecture, it can be seen from (2.4a-2.4b) that this additional error is given by $E[\Delta c^T(n)\mathbf{R}\Delta c(n) + \Delta d^T(n)\mathbf{R}\Delta d(n)]$, where $\Delta c(n)$ and $\Delta d(n)$ are the errors due to quantization of coefficients $c(n)$ and $d(n)$, respectively and $\mathbf{R} = E[\mathbf{X}(n)\mathbf{X}^H(n)]$ is the correlation matrix of the input signal. Next, by assuming a uniform noise model for the quantization error, and $\sigma_{F,CC}^2 = 2^{-2B_{F,CC}}/12$, it can be seen that the quantization error is given by $2N\sigma_{F,CC}^2\sigma_x^2$. Therefore, if J_o is the specified *MSE*, the **F**-block precision is given by,

$$B_{F,CC} > \frac{1}{2} \log_2 \left(\frac{N\sigma_x^2}{6(J_o - J_{float})} \right). \quad (3.1)$$

Similarly, the **F**-block precision for the **SR** architecture can be determined. By collecting all error terms due to coefficient quantization in (2.5), It can be easily shown that the error due to fixed-point implementation of the **F**-block in strength-reduced architecture is given by $3N\sigma_{F,SR}^2\sigma_x^2$. Therefore, **F**-block precision, $B_{F,SR}$ is given by,

$$B_{F,SR} > \frac{1}{2} \log_2 \left(\frac{N\sigma_x^2}{4(J_o - J_{float})} \right). \quad (3.2)$$

Thus it can be seen that the coefficient precision of **F**-block for **CC** and **SR** architectures is related by,

$$B_{F,SR} = B_{F,CC} + 0.3. \quad (3.3)$$

This shows that the **F**-block in the **SR** architecture requires at the most one bit more than in the **CC** architecture. The **F**-block precision requirements for **PIPSR** architecture (see (2.7)) is same as that of the **SR** architecture, because both architectures involve same computations in the **F**-block.

B. WUD-block Precision

The finite precision **WUD** block can be analyzed by using linear model for coefficient quantization noise. Then, B_{WUD} is chosen based on the *stopping criterion* (see 1.1)

[2]. The precision assigned should be sufficient for the adaptive filter to converge to the specified *MSE*, J_o .

For **CC** architecture, the correction terms are given by (2.4c-2.4d). Using the stochastic estimates for these terms and on applying stopping criterion we get,

$$B_{WUD,CC} \geq \frac{1}{2} \log_2 \left(\frac{2}{\mu^2 J_o \sigma^2} \right), \quad (3.4)$$

where J_o is the desired *MSE* level.

A similar expression can be found for the coefficient precision of the **WUD**-block in the **SR** architecture. If we use stochastic estimates $e\mathbf{X}_1(n)$, $e\mathbf{X}_2(n)$ and $e\mathbf{X}_3(n)$ in (2.6), the coefficient precision of the **WUD** block in **SR** architecture is given by,

$$B_{WUD,SR} \geq \frac{1}{2} \log_2 \left(\frac{2}{\mu^2 J_o \sigma^2} \right) - 0.5. \quad (3.5)$$

Comparing (3.4) and (3.5), we see that the precision requirements for **WUD**-block in the **SR** architecture are 0.5 bits less than that of the **CC** architecture. This is an advantage of the **SR** architecture over the **CC** architecture. This is indeed an attractive result given that the **SR** architecture also enables power savings of 21 – 25% [6].

The precision requirements for **WUD** block of **PIPSR** architecture (see 2.8) can be determined by replacing μ in the above analysis by μLA .

IV. APPLICATION TO 155.52Mb/s ATM-LAN

In this section, we employ the finite precision **PIPSR** architecture as a NEXT canceller for 155.52Mb/s ATM-LAN [5] over UTP-3 cable and present the simulation results.

A. 155.52Mb/s ATM-LAN Transceiver

The basic transceiver block diagram is presented in Fig. 2. The transmitter consists of a 64-CAP (6 bits/symbol) encoder and shaping filters with sampling rate of 77.76Msamples/s, excess bandwidth $\alpha = 15\%$ and span of 8 symbol periods. At the receiver (see Fig. 2), the received signal is distorted further due to the superimposition of the NEXT signal. This composite signal is processed by a fractionally space linear equalizer (FSLE), which is a pair of adaptive filters. In addition, the local transmitted symbols are passed through a complex adaptive NEXT canceller, which tries to cancel the effect of NEXT in the received signal. We employ the finite-precision architectures presented in this paper as NEXT cancellers. We will assume that **PIPSR** NEXT canceller has been obtained by pipelining **SR** architecture to the pipelining level of 105 by using $D_1 = 109$, $D_2 = 5$ and $LA = 2$ (see [6] for more details regarding this choice of D_1 , D_2 and LA). For floating point algorithm, J_{float} is 0.0435, which corresponds to SNR_o (defined as

$42/J_{float}$ for 64-CAP) of 29.85dB. Desired SNR_o (corresponding to probability of error of 10^{-10}) is 29.75dB (or $J_o = 0.0445$). For NEXT canceller being considered, $\sigma_x^2 = 42$.

B. Simulation Results

F-block precisions can be determined by employing (3.1) for CC architecture and (3.2) for both SR and PIPSR architectures. On substituting above given parameters, we obtain $B_{F,CC} = 8.87$, $B_{F,SR} = 9.17$ and $B_{F,PIPSR} = 9.17$. This is also confirmed by simulation results plotted in Fig. 3, which shows variation of the SNR_{slicer} with the F-block precision in CC, and SR architectures. Desired SNR is attained at about 9 bit precision for CC architecture and 10 bits for SR architecture. Fig. 3 also shows that coefficient precision required in F-block for the SR architecture is at the most 1 bit more as compared to the CC architecture. Recall that this conclusion was also obtained from (3.3).

Similarly, the coefficient precision in the WUD-block can be determined by employing (3.4) for CC, (3.5) for SR and (3.5) with μ replaced by μLA for the PIPSR. For proper convergence, μ was chosen to be 0.0007, 0.0007 and 0.0002 for CC, SR and PIPSR algorithms respectively. B_{WUD} precisions are then determined (Section III) to be $B_{WUD,CC} = 9.45$, $B_{WUD,SR} = 8.95$ and $B_{WUD,PIPSR} = 9.51$. These results are confirmed by simulation results in Fig. 4, where desired performance is reached for 9 bit precision for both CC and SR architectures.

Therefore, we conclude that CC and SR architectures have similar coefficient precision requirements.

REFERENCES

- [1] N. J. Bershad and J. C. M. Bermudez, "A nonlinear analytical model for the quantized LMS algorithm - the powers-of-two case," *IEEE Trans. Signal Processing*, vol. 44, no. 11, pp. 2895-2900, Nov 1996.
- [2] C. Caraiscos and B. Liu, "A roundoff error analysis of the LMS adaptive algorithm," *IEEE Trans. Acoust., Speech, and Signal Process.*, vol. ASSP-32, no. 1, pp. 34-41, Feb 1984.
- [3] A. Chandrakasan et al., "Minimizing power using transformations," *IEEE Trans. Comp.-Aided Design*, vol. 14, no. 1, pp. 12-31, Jan. 1995.
- [4] R. D. Gitlin, J. F. Hayes, and S. B. Weinstein, *Data Communications Principles*. NY: Plenum Press, 1992.
- [5] G. H. Im and J. J. Werner, "Bandwidth-efficient digital transmission up to 155 Mb/s over unshielded twisted-pair wiring," *IEEE Conf. on Comm.*, vol. 3, pp. 1797-1803, May 1993.
- [6] N. R. Shanbhag and M. Goel, "Low-power adaptive filter architectures and their application to 51.84 Mb/s ATM-LAN," *IEEE Trans. Signal Processing*, vol. 45, no. 5, pp. 1276-1290, May 1997.
- [7] N. R. Shanbhag and K. K. Parhi, "Relaxed look-ahead pipelined LMS adaptive filters and their application to ADPCM coder," *IEEE Trans. Circuits and Systems*, vol. 40, pp. 753-766, Dec. 1993.
- [8] B. Widrow et al., "Stationary and non-stationary learning characteristics of the LMS adaptive filter," *Proc. IEEE*, vol. 1964, pp. 1151-1162, Aug. 1976.

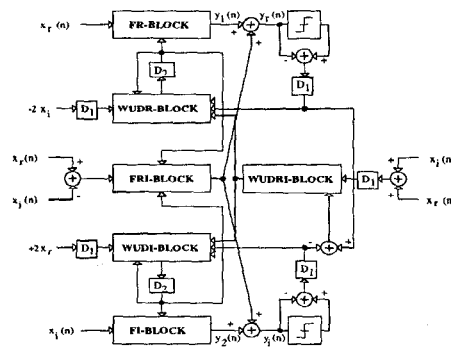


Fig. 1. PIPSR Architecture

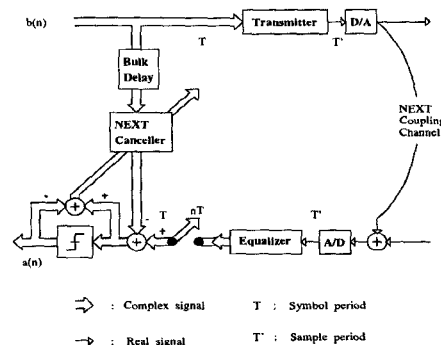


Fig. 2. 155.52Mb/s ATM-LAN Transceiver

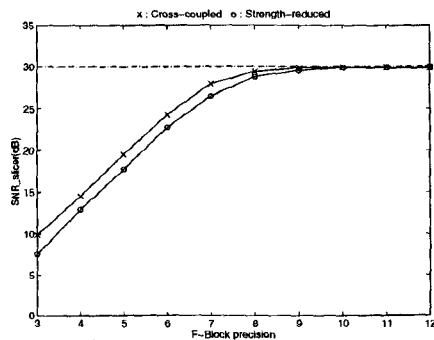


Fig. 3. F-block precision

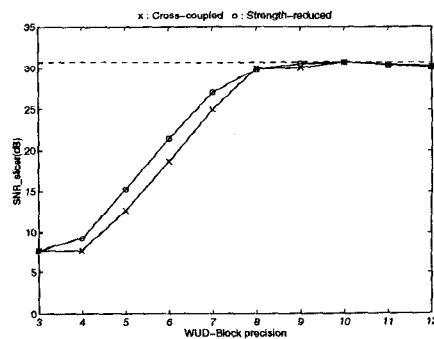


Fig. 4. WUD-block precision