High-Performance VLSI Architecture of DLMS Adaptive Filter for Fast-Convergence and Low-MSE

Mohd. Tasleem Khan and Rafi Ahamed Shaik, *Member, IEEE*

***Abstract*—This brief presents a high-performance VLSI architecture of delayed least mean square (DLMS) adaptive filter for fast-convergence and low-mean square error (MSE) using distributed arithmetic (DA). The proposed design estimates response against the adaptation delays using a parallel predictive adder tree followed by a shift accumulate (SA) unit. An efficient quantization scheme with two bits of scaled error signal is also suggested. Single SA unit for multiple DA bases is used to reduce the number of adders and registers. Simulation and synthesis results show that the proposed design for 32nd order provides 19.72% lesser area, 25.51% lesser power, lesser 28.89% MSE and 59.91% lesser MSE/area over the best existing design.**

***Index Terms*—Adder tree (AT), adaptive filter (ADF), least mean square (LMS), mean square error (MSE), pipelining.**

I. Introduction

ADAPTIVE filter (ADF) using least-mean-square (LMS) algorithm is widely used in channel equalization to cancel inter-symbol interference [1]. Next generation 5G communication demands low-complexity, high SNR and high throughput systems [2]. Obtaining high-throughput and low-complexity ADF is achievable with pipelining at the cost of SNR [3]. However, it is difficult to pipeline LMS ADF due to feedback loop in the coefficients adaptation. To overcome this issue, delayed LMS (DLMS) algorithm using adaptation delays has been suggested [4]. As the number of adaptation delays increases, it slows down the convergence rate and increases mean-square error (MSE). Further, the accumulation of quantization errors deteriorate the MSE [5], [6].

Distributed arithmetic (DA) [7] is a popular multiplier-less approach for efficient realization of LMS ADF [8]–[14]. It is a bit-serial approach that relies on two's complement (TC), which has a look-up table (LUT) and a shift-accumulate (SA) unit. LUT stores the partial products with addresses as bit-slice coefficients, and SA unit produces the output. In TC-DA, the LUT size varies exponentially with number of taps. To avoid this limitation, researchers have used either two LUTs [8] or one LUT [9]. In [10], a new high throughput design for LMS ADF using offset-binary-coding (OBC) has been reported. Due to inherent symmetries in OBC-DA, its eigenvalue spread of input correlation matrix [15] is always higher than TC-DA. Consequently, it reduces the convergence rate and increases the MSE of LMS ADF [16]. There was a paradigm shift from LMS to DLMS based on TC-DA for high throughput realization [11]. In [13], we had presented three new OBC-DA based DLMS ADF designs (Structure-I, II, III) based on the framework of [11]. These structures were hardware optimized using serial LUTs at the expense of undesired non-OBC terms ($\sum_{t=0}^{m} x_t$, where $x_{t|t\in[0,m]}$ are first $(m+1)$ samples, and m denotes the adaptation delays). They were generated due to the presence of all 0's present in LUT addresses for $(m+1)$ clock cycles. The clock period of these designs were reduced using carry-save adders at expense of an extra adder tree. In [12], a strategy was presented to improve the MSE of DLMS ADF based on approximations of [17]. Some recent works on DLMS ADF using multipliers have been reported [18]–[20].

From VLSI implementation point-of-view, Structure-III [13] could be a good choice for low-cost, but it possess symmetries and undesired terms. While TC-DA is beneficial for DLMS ADF in the sense that bit-slices do not exhibit symmetries, and offers low-cost for serial-LUTs with AND gates. However, the removal of undesired terms with further reduction in hardware cost and error analysis are yet to be addressed. This motivates us to develop high-performance architecture of DLMS ADF. The key contributions of the present work are as follows:

- Formulation of the DLMS ADF based on TC-DA.
- Parallel predictive adder tree (PPAT) followed by a SA unit to estimate the response against adaptation delays.
- The number of adders/registers are reduced by sharing a single SA unit for multiple DA bases.
- Efficient quantization scheme with two-bits of scaled error followed by error analysis based on signal statistics.
- Extensive simulations are performed to evaluate the performance of proposed design for band limited channels.

The rest of the brief is organized as follows: We formulate the proposed design in Section II. Next, we present architectural details and error analysis in Sections III and IV respectively. Lastly, we evaluate and compare the performance of proposed design using application synthesis integrated circuit (ASIC) in Section V. Conclusion is provided in Section VI.

II. Mathematical Formulation

At any iteration n, the output y_n of a N-tap FIR filter with input $\mathbf{x}_n = \{x_{n-i}\}_{i=0}^{N-1}$ is given by

$$y_n = \mathbf{w}_n^T \mathbf{x}_n \tag{1}$$

where $\mathbf{w}_n = \{w_n(i)\}_{i=0}^{N-1}$ are the filter coefficients. An error e_n is computed by subtracting y_n from the desired signal d_n as $e_n = d_n - y_n$. Based on e_n, the coefficients are updated using LMS criterion as

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu e_n \mathbf{x}_n \tag{2}$$

Manuscript received December 8, 2021; revised December 26, 2021; accepted January 6, 2022. Date of publication January 10, 2022; date of current version March 28, 2022. This brief was recommended by Associate Editor G. Jovanovic Dolecek. *(Corresponding author: Mohd. Tasleem Khan.)*

Mohd. Tasleem Khan is with the Department of Electronics Engineering, Indian Institute of Technology Dhanbad, Dhanbad 826004, India (e-mail: mtkhan@iitism.ac.in).

Rafi Ahamed Shaik is with the Electrical and Electronics Engineering Department, Indian Institute of Technology Guwahati, Guwahati 781039, India.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSII.2022.3141687.

Digital Object Identifier 10.1109/TCSII.2022.3141687

where μ is the step size. For pipelined ADF, DLMS algorithm is suggested [4]. Using bit level transformation of coefficient update, DLMS algorithm can be written as

$$\mathbf{w}_{l+1}(i) = \mathbf{w}_l(i) + D_l\{\Delta \mathbf{w}_{n-m}\} \tag{3}$$

where $\Delta \mathbf{w}_{n-m} = \mu e_{n-m}\mathbf{x}_{n-m}$, $\mathbf{w}_l(i)(i = 0, 1, \ldots, N-1)$ is the bit-slice coefficient vector at l^{th} clock cycle, $l \in [0, B-1]$, B is the coefficient wordlength and $D_l\{\cdot\}$ is the unit delay operator. The coefficients $w_n(i)$ in TC form can be given as

$$w_n(i) = \mathbf{w}_n^T(i)\mathbf{s} \tag{4}$$

where $\mathbf{w}_n(i) = \{w_n(i, l)\}_{l=0}^{B-1}$, $\mathbf{s} = \{(-1)^r 2^{-l}\}_{l=0}^{B-1}$ and $r = \lfloor (B-1-l)/(B-1) \rfloor$. By substituting Eq. (4) in Eq. (1) and interchanging the order of summation, we obtain

$$y_n = \mathbf{c}_n^T\mathbf{s} \;\&\; \mathbf{c}_n = \mathbf{w}_n^T(i)\mathbf{x}_n = \{c_n(l)\}_{l=0}^{B-1} \tag{5}$$

Each $c_n(l)$ would take 2^N combinations of input samples which are shift-and-accumulated for B clock cycles to produce y_n, as per Eq. (5). These combinations can be stored in a LUT with coefficients bit-slice as its address lines. With decomposition $N = L \times M$, LUT can be split into M smaller LUTs of each size 2^L, where $c_n(l)$ can be written as

$$c_n(l) = \sum_{j=0}^{M-1} \mathbf{w}_n^T(j)\mathbf{x}_n(j) \tag{6}$$

where $\mathbf{w}_n(j) = \{w_n(k + jL, l)\}_{k=0}^{L-1}$ with $j \in [0, M-1]$. From Eq. (6), it can be noted that M inner products of size L are added using an adder-tree (AT) of $\log_2 M$ depth.

The output of DLMS ADF using Eq. (5) and Eq. (6) with first m and last $(L-m)$ coefficients can be expressed as

$$y_{n-m} = \left[\sum_{j=0}^{M-1} \mathbf{w}_n^m(j)\mathbf{x}_n^m(j) + \mathbf{w}_n^{L-m}(j)\mathbf{x}_n^{L-m}(j)\right]^T \mathbf{s} \tag{7}$$

where $\mathbf{w}_n^m(j) = \{w_n(k_1 + jL, l)\}_{k_1=0}^{m-1}$, $\mathbf{x}_n^m(j) = \mathbf{x}_{n-m}(j) = \{x_{n-k_1-jL}\}_{k_1=0}^{m-1}$, $\mathbf{w}_n^{L-m}(j) = \{w_n(k_2+jL, l)\}_{k_2=m}^{L-1}$, and $\mathbf{x}_n^{L-m}(j) = \mathbf{x}_{n-jL+m} = \{x_{n-k_2-jL}\}_{k_2=m}^{L-1}$. Due to '$m$' adaptation delays, the computation of e_{n-m} can be considered as the constraint optimization problem with constraint

$$\text{Min}\left\{\mathbb{E}|d_{n-m} - y_{n-m}|^2\right\}, \text{s.t.}\mathbf{w}_n^m(j) = \mathbf{w}_o^m(j) \tag{8}$$

where $\mathbf{w}_o^m(j)$ denotes the m-optimal coefficients and $\mathbb{E}(\cdot)$ is an expectation operator.

III. Proposed Architecture

Based on the above derivation, we proposed a scalable architecture for DLMS ADF which can be extended to different N, L and m. The proposed design for $N = 16$, $L = 4$ and $m = 2$ is shown in Fig. 1. It consists of 4 LUTs with 4 upper ($U_{j|j\in\{0,3\}}$)-LUTs and 4 lower ($L_{j|j\in\{0,3\}}$)-LUTs components, 2 ATs, 2 SA units, an error computation unit (ECU), 4 bit-level coefficient update units (CUUs), and a control unit. The LUTs are realized serially with AND gates followed by adders whose outputs are added with separate ATs. Precisely, one AT adds the outputs of U_j/L_j-LUTs followed by SA unit to produce y_{n-m}, as per Eq. (7). While other AT only adds the outputs of U_j-LUT followed by another SA unit. The AT of depth $\log_2 M$ which adds U_j-LUT outputs is referred to as PPAT.

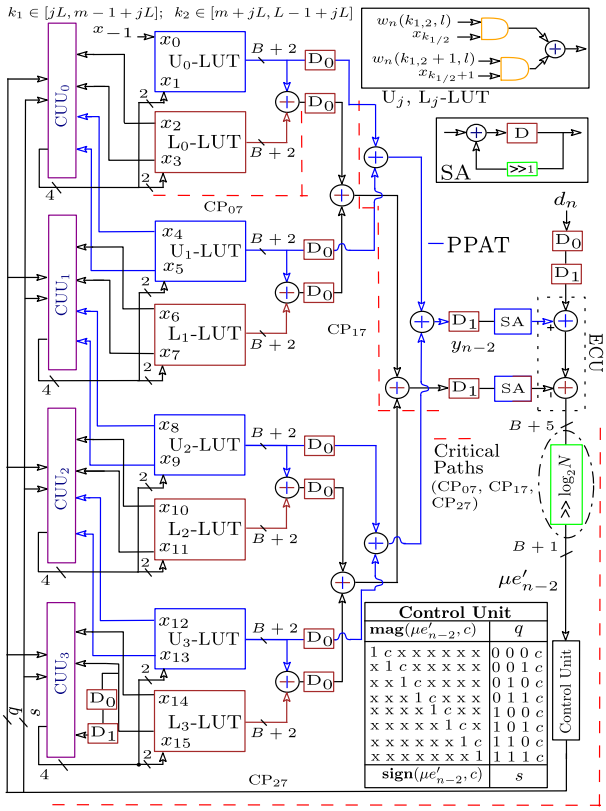

Fig. 1. PPAT-DLMS ADF architecture for $N = 16$, $L = 4$ and $m = 2$.

The bit-level CUU is implemented, as per Eq. (3), with slight modification in barrel-shifters (BSs) as compared to [11], [13]. Note that the past input samples $\mathbf{x}_n^m(j)$ for each serial-LUT are not available. But, they are used to pipeline the architecture with D_0 and D_1 adaptation delays for $m = 2$, as shown in Fig. 1. The U_j-LUT and L_j-LUT outputs of a given j^{th}-serial LUT are produced with inputs $\{x_{0+4j}, x_{1+4j}\}$ and $\{x_{2+4j}, x_{3+4j}\}$ respectively. Unlike [11], the adaptation delays are retimed, and chosen to be placed after first-level of AT and before ECU to minimize the critical-path (CP) delays. Observably, the locations of D_0 and D_1 for both the ATs are same in order to ensure their correct arrival, and subsequent removal at ECU. Note that D_0 and D_1 affect the first two coefficients of each serial-LUT. After ECU, e_{n-2} is scaled by μ to obtain the scaled error μe_{n-2}, as per Eq. (3). Unlike [11], [13], an additional-bit through sign-extension is considered to add the outputs from U_j/L_j-LUTs. By doing so, the multiplication of μ with e_{n-2} would truncate $(\log_2 N)$-bits from e_{n-2}, and leave $(B+1)$-bits in scaled error μe_{n-2}. As a consequence, its most-significant bit (MSB) followed by next-bit (c) can be used to compute $\mu e_{n-2}\mathbf{x}_n(j)$ in CUU, as shown by control unit in Fig. 1. Depending upon c, an extra level of 2-to-1 multiplexers in BSs would be required to allow inputs as $\mathbf{x}_n(j)$ or $2^{-1}\mathbf{x}_n(j)$. The most-significant bit and c-bit of $(\mu e_{n-2}, c)$ are used to generate $q(c)$ selection lines for 2-to-1 multiplexers in BSs, and sign of $(\mu e_{n-2}, c)$ is used to generate s-bit to update the coefficients.

IV. Quantization and Error Analysis

For fixed-point (FP) realization, it is required to obtain fractional and integer-parts of wordlength for desired accuracy.

Suppose (X, X_i) is the FP representation of $\mathbf{w}_n$ and $\mathbf{x}_n$ with $X = B, X_i = B_i$ and $X = W, X_i = W_i$ respectively. The FP representation of y_{n-m} is $(B+W+\log_2 N+1, B_i+W_i+\log_2 N+1)$, where $\log_2 N + 1$ arises due to increase in bits through serial LUTs and ATs. In the proposed design, y_{n-m} is required to have $(B + \log_2 N + 1, B_i + W_i + \log_2 N + 1)$ bits. As PPAT-DLMS ADF performs learning, the sign of d_{n-m} has to be same as y_{n-m}, while its size can be made equal to y_{n-m} through specific-scaling. This implies that e_{n-m} could have the same representation as that of y_{n-m}. In [4], the convergence of DLMS ADF is guaranteed if

$$0 < \mu < 2/(\sigma_x^2(N-2) + 2m - 2)\sigma_x^2 \tag{9}$$

where σ_x^2 is the average power of input samples. Consider μ is represented as 2^{-t} with $t \in (\log_2 N, B_i + W_i + \log_2 N + 1)$. If $t \rightarrow B_i + W_i + \log_2 N + 1$, then it does not lead to truncation since multiplication of μ with e_{n-m} only changes the radix-point. However, if we need smaller μ, i.e., $t \rightarrow B_i + W_i + 2 \cdot \log_2 N + 1$, then it truncates $(\log_2 N)$-bits from e_{n-m} due to c-bit. As per Eq. (3), $\Delta\mathbf{w}_{n-m}$ must have the representation $(B + W, B_i + W_i)$. As $\mathbf{w}_n$ converge toward the optimal value, $\Delta\mathbf{w}_{n-m}$ become smaller, and contains zeros at the MSB side. Further, $(B-B_i)$ LSBs of $\Delta\mathbf{w}_{n-m}$ are truncated to have same FP representation as $\mathbf{w}_n$. From above discussion, $\mathbf{x}_{n-m}$, d_{n-m}, $\mathbf{w}_{n-m}$, and y_{n-m} in FP can be expressed as $\mathbf{x'}_{n-m} = \mathbf{x}_{n-m} + \boldsymbol{\epsilon}_x$, $d'_{n-m} = d_{n-m} + \epsilon_d$, $\mathbf{w'}_{n-m} = \mathbf{w}_{n-m} + \boldsymbol{\epsilon}_w$, and $y'_{n-m} = y_{n-m} + \epsilon_y$ respectively, where $\boldsymbol{\epsilon}_x/\epsilon_d/\boldsymbol{\epsilon}_w/\epsilon_y$ are their associated independent quantization errors [5]. Substituting y'_{n-m} in Eq. (8) leads to

$$\text{Min}\left\{\mathbb{E}|d_{n-m} - y'_{n-m}|^2\right\} = \text{Min}\left\{\mathbb{E}|e_{n-m}|^2 + \sum \mathbb{E}|f(\epsilon)|^2\right\}, \tag{10}$$

where $\sum \mathbb{E}|f(\epsilon)|^2$ represents the composite effect of quantization errors $\boldsymbol{\epsilon}_x/\epsilon_d/\boldsymbol{\epsilon}_w/\epsilon_y$. In TC-form, the first m and last $(L-m)$-optimal coefficients of conventional LMS and DLMS ADFs are expressed as $\mathbf{w}_o = [\mathbf{w}_o^m \mathbf{s} \mathbf{w}_o^{L-m} \mathbf{s}]$ and $\mathbf{w}_d = [\mathbf{w}_d^m \mathbf{s} \mathbf{w}_d^{L-m} \mathbf{s}]$ respectively. The MSE of PPAT-DLMS ADF for given m coefficients $\mathbf{w} = [\mathbf{w}_n^m \mathbf{s} \mathbf{w}_n^{L-m} \mathbf{s}]$ can be expressed as

$$\begin{aligned}
\xi_d(\mathbf{w}) &= \xi_{0,o} + (\mathbf{w} - \mathbf{w}_o)^T \mathbf{R}_o (\mathbf{w} - \mathbf{w}_o) = \xi_{0,o} + \delta\mathbf{w}^T \mathbf{R}_o \delta\mathbf{w} \\
&= \xi_{0,o} + \delta\mathbf{w}^T \begin{bmatrix} \mathbf{R}_{xx} & \mathbf{P}_{xx^m} & \mathbf{P}_{xx^{L-m}} \\ \mathbf{P}_{xx^m}^T & \sigma_t^2\mathbf{I} & 0 \\ \mathbf{P}_{xx^{L-m}}^T & 0 & \sigma_t^2\mathbf{I} \end{bmatrix} \delta\mathbf{w} \\
&= \xi_{0,o} + \mathbf{a}^T \mathbf{R}_d \mathbf{a} + \mathbf{b}^T \mathbf{Q}_{xx^m} \mathbf{a} + \mathbf{a}\mathbf{Q}_{xx^m}^T \mathbf{b} + \sigma_t^2 \mathbf{b}^T \mathbf{b}
\end{aligned} \tag{11}$$

where $\xi_{0,o}$ is the MSE of TC-DA based LMS ADFs [8], [9], σ_t^2 is the variance of transmitted data, $\mathbf{R}_d = \mathbb{E}\{\mathbf{x}_{n-jm}\mathbf{x}_{n-jm}^T\}$ is the auto-correlation of PPAT-DLMS ADF, $\mathbf{R}_{xx} = \mathbb{E}\{\mathbf{x}_{n-jL+m}\mathbf{x}_{n-jL+m}^T\}$ is the auto-correlation of input samples, $\mathbf{P}_{xx^m} = \mathbb{E}\{\mathbf{x}_{n-jL}\mathbf{x}_{n-jm}^T\}$ is the cross-correlation of $\mathbf{x}_{n-jL}$ and $\mathbf{x}_{n-jm}$, $\mathbf{P}_{xx^{L-m}} = \mathbb{E}\{\mathbf{x}_{n-jL}\mathbf{x}_{n-j(L-m)}^T\}$ is the cross-correlation of $\mathbf{x}_{n-jL}$ and $\mathbf{x}_{n-j(L-m)}$, $\mathbf{Q}_{xx^m} \triangleq \left[\mathbf{P}_{xx^m}^T\ 0\right]$, $\mathbf{a} = \mathbf{w}_n^{L-m}\mathbf{s} - \mathbf{w}_o^{L-m}\mathbf{s}$, $\mathbf{b} = \mathbf{w}_d^m\mathbf{s} - \mathbf{w}_o^m\mathbf{s}$. Since $\mathbf{R}_d$ is a positive-definite matrix, Cholesky factorization can be applied as $\mathbf{R}_d = \mathbb{R}\mathbb{R}^T$, (where $\mathbb{R}$ is the lower-triangular matrix). Thus, Eq. (10) becomes

$$\begin{aligned}
\xi_d(\mathbf{w}) &= \xi_{0,o} + (\mathbf{a}^T\mathbb{R} + \mathbf{b}^T\mathbf{Q}_{xx^m}(\mathbb{R}^T)^{-1}) \times (\mathbb{R}^T\mathbf{a} + \\
&\quad \mathbb{R}^{-1}\mathbf{Q}_{xx^m}^T\mathbf{b}) - \mathbf{b}^T\mathbf{Q}_{xx^m} \times \mathbf{R}_d^{-1}\mathbf{Q}_{xx^m}^T\mathbf{b} + \sigma_t^2\mathbf{b}^T\mathbf{b} \\
&= \xi_{0,d} + \mathbf{U}^T\mathbf{U} + \mathbf{b}^T \times (\sigma_t^2\mathbf{I} - \mathbf{Q}_{xx^m} \times \mathbf{R}_d^{-1}\mathbf{Q}_{xx^m}^T)\mathbf{b} \\
&= \xi_{0,d} + \mathbf{U}^T\mathbf{U} + \mathbf{b}^T\mathbf{V}\mathbf{b}
\end{aligned} \tag{12}$$

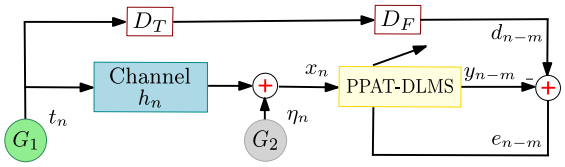

Fig. 2. Block diagram to evaluate PPAT DLMS ADF in channel equalization.

where $\mathbf{U} \triangleq \mathbb{R}^T\mathbf{a} - \mathbb{R}^{-1}\mathbf{Q}_{xx^m}^T$, $\mathbf{V} \triangleq \sigma_t^2\mathbf{I} - \mathbf{Q}_{xx^m}\mathbf{R}_d^{-1}\mathbf{Q}_{xx^m}^T$. Since $\mathbf{U}^T\mathbf{U} \geq 0$, we can obtain $\xi_{d,0}(\mathbf{w}) = \xi_d(\mathbf{U}=\mathbf{0}) = \xi_{0,o} + \tilde{\xi}$, where $\tilde{\xi}$ denotes the increase in MSE of PPAT-DLMS.

V. Results and Discussion

Fig. 2 shows block-diagram to evaluate PPAT-DLMS ADF in channel equalization with bandlimited channel [21] as

$$h_n = 0.5 * [1 + \cos(2\pi(n-2)/P)] \tag{13}$$

for $n = 1, 2, 3$ otherwise $h_n = 0$; where P is the amplitude distortion introduced by channel. Equivalently, it controls the eigen-value spread (or condition number $\chi(\mathbf{R})$ [15]). Two independent random number generators (G_1, G_2) are considered as shown in Fig. 2, where G_1 provides the test signal t_n to probe the channel with random sequence in polar form ± 1 i.e., zero-mean and unit variance and G_2 serves as the source of AWGN (η_n) to corrupt the channel output. Note that desired signal d_{n-m} is a delayed version of t_n, where delays are due to channel (D_T) and PPAT-DLMS ADF (D_F). As per Eq. (13), h_n is symmetric about $n = 2$ which implies $D_T = 2$, and direct-form FIR filter with m delays in PPAT-DLMS ADF results $D_F = \lfloor N/2 \rfloor + m$. Thus, t_n is to be delayed by $(2 + \lfloor N/2 \rfloor + m)$ clock cycles in order to feed d_{n-m}. To validate the proposed architecture, we consider TC-DA based LMS [8], [9] and DLMS [11], [12] ADFs as the benchmarks in the following experiments.

Experiment-1: Effect of $\chi(\mathbf{R})$. The PPAT-DLMS/DLMS/LMS ADFs based on TC-DA are simulated for $\chi(\mathbf{R}) = 105.4$ (at $P = 3.62$) and $\chi(\mathbf{R}) = 6.6$ (at $P = 2.91$) with $m = 2$, $L = 4$, $N = 16$, $B = W = 8$, $B_i = 2$, $W_i = 0$ and SNR= 40 dB. The first-tap input at time-instant n is given by $x_n = \sum_{k=1}^{3} h_k t_{n-k} + \eta_n$. It implies that $(N+m)$-taps input correlation matrix $\mathbf{R}_{xx}$ for PPAT-DLMS ADF is a symmetric matrix. Since $h_n \neq 0$ for $n \in [1, 3]$, and η_n is a white-noise with zero mean and variance σ_η^2, the elements of $\mathbf{R}_{xx}$ is denoted by $[r_{ii}]_{i|i\in[1,N+m]}$ with $r_{ii} = h_1^2 + h_2^2 + h_3^2 + \sigma_\eta^2$; $r_{i2} = r_{2i} = h_1h_2 + h_2h_3$; $r_{i3} = r_{3i} = h_1h_3$; and $r_{iq} = r_{qi} = 0$; $\forall q \in [4, N+m]$. The simulations were performed by averaging the squared error over 100 independent trials with $\mu = 0.03125$, as shown in Fig. 3(a). Clearly, increasing $\chi(\mathbf{R})$ has an adverse effect on convergence rate and MSE as their inputs are ill-conditioned. This effect is more pronounced for the case of DLMS ADF. This is because adaptation delays causing the inputs to be more ill-conditioned. In contrast, PPAT-DLMS ADF can overcome this issue and provide 8.54 dB reduction in MSE with almost 197 less iterations at $\chi(\mathbf{R}) = 105.4$ than DLMS ADF to reach the steady-state, and almost matches with LMS ADF.

Experiment-2: Effect of wordlengths B, W. In this experiment, PPAT-DLMS/DLMS/LMS ADFs are considered for $B, W = 6, 4$ and $16, 12$ with $P = 2.91$, $m = 4$, $L = 8$, $N = 32$, $B_i = W_i = 0$ and input SNR= 30 dB, as shown in Fig. 3(b). Clearly, the convergence rate with $m = 4$ worsens further apart

TABLE I
GENERAL COMPARISON OF HARDWARE AND TIME COMPLEXITIES OF DIFFERENT DA BASED ADF ARCHITECTURES

Design	ADD	REG	MUX	BS	CP	NCC	LUT
†DA_1 [9]	$(L+2)M+1$	$(3L+2)M+1$	γ_0	LM	CP_{01}	n_1	$2^L M$
*DA_2 [10]	$4M+2$	$(4+L)M+2$	$2M+\gamma_1$	M	CP_{02}	n_2	$4\cdot 2^{L-2}M$
‡DA_3 [11]	$(2+2^{L-1})M+LM+\alpha_0$	$(1+m+2^L)M+2N+2m+\beta_0$	$2^L M+\gamma_2$	LM	$\max[CP_{03},...,CP_{(m-1)3}]$	1	–
‡DA_4 [12]	$(3+2^{L-2})M+N'+\alpha_0$	$(2+m+2^{L-1})M+2N'+N+2m+\beta_0$	$(2^{L-1}+1)M+\gamma_3$	LM	$\max[CP_{04},...,CP_{(m-1)4}]$	1	–
**DA_5^1 [13]	$(2L+3)M+N'+1+\alpha_1$	$(2L+4)M+2N'+3+2m+\beta_1$	$M+1+\gamma_{4,5}$	LM	$\max[CP_{05}^1,...,CP_{(m)5}^1]$	n_3	–
**$DA_5^{2,3}$ [13]	$(L+3)M+N'+1+\alpha_{2,3}$	$(2L+1+1.5m)M+2N'+3+2m+\beta_{2,3}$	$M+1+\gamma_6$	LM	$\max[CP_{05}^{2,3},...,CP_{(m)5}^{2,3}]$	1	–
*DA_6 [14]	$6M+2$	$(3+L)M+2$	$4M+\gamma_7$	M	CP_{06}	n_4	$2\cdot 2^{L-1}M$
‡Proposed	$(L+1)M+N'+2+\alpha_4$	$(2L+m)M+2N'+3m+\beta_4$	–	$LM+N''$	$\max[CP_{07},...,CP_{(m)7}]$	1	–

†/‡: TC-based LMS/DLMS ADFs, */**: OBC-based LMS/DLMS ADFs, **ADD**: Adders, **REG**: Registers, **MUX**: Multiplexers, **BS**: barrel shifters, **CP**: critical-path and **NCC**: number of clock cycles, where $N = L \times M$, $N' = N/B$, $N'' = N'(W+1)$ (additional level of 2-to-1 multiplexers in BS), and W/B are wordlengths of inputs/coefficients, respectively. Note that bit-level complexities (α_k, β_k, γ_k), CPs (with $m=2$) and NCC (n_k) of DA_{1-5} are listed at the footnote of Table III [13], where k indicates the design. The bit-level complexities of the proposed design are $\alpha_4 = M(L(\log_2 L+1)/4+1)+2(\log_2 N-2)$, $\beta_4 = 2(L-1)M+2(\log_2 N-2)$; and the CPs with $m=2$ are $CP_{07} = T_{AND}+(\log_2 L+1)T_A+T_X+T_D$, $CP_{17} = (\log_2 M+1)T_A+T_X+T_D$ and $CP_{27} = 2T_A$ where T_A, T_X and T_D are delays adder, XOR gate and register, respectively.

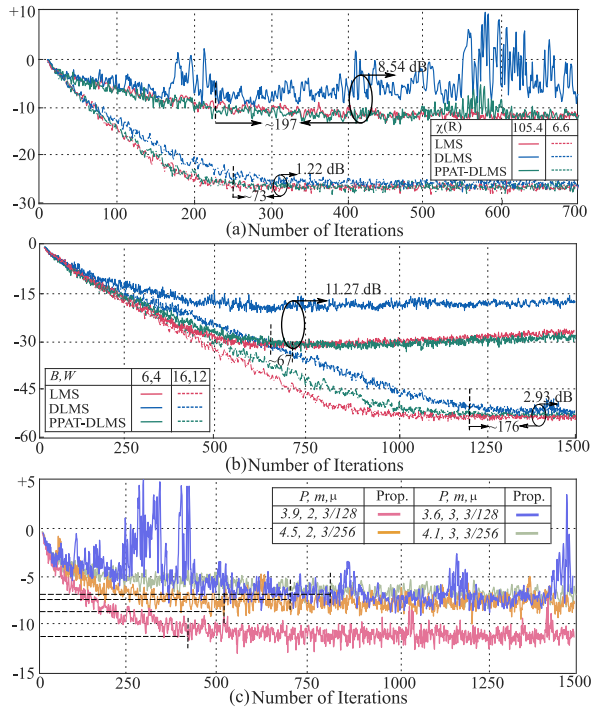

Fig. 3. Comparison of convergence rate and MSE for (a) Experiment-1 ($\chi(\mathbf{R})$ effect) (b) Experiment-2 (B, W effect), and (c) Experiment-3 (P, m effect).

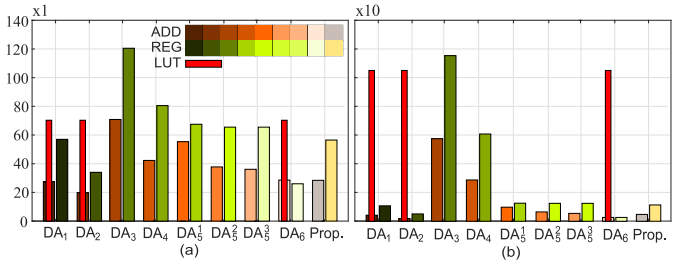

Fig. 4. ADD/REG of Proposed/DA_{1-6} for N, L, m (a) 16, 4, 2; (b) 32, 8, 4.

from large P values. Compared to DLMS ADF, the PPAT-DLMS ADF for $B, W = 16, 12$ achieves the steady-state with almost 176 less iterations and 2.93 dB less MSE. However, in case of $B, W = 6, 4$, the steady-state MSEs of all ADFs are increased due to quantization with shorter wordlength, as per Eq. (10). Note that ill-effect of shorter wordlengths on MSE is more enhanced for DLMS ADF. For instance, DLMS ADF with $B, W = 6, 4$ needs almost 67 more iterations, and 11.27 dB more MSE to reach steady-state than PPAT-DLMS ADF. The proposed design has slightly better MSE than LMS ADF due to additional c-bit.

Experiment-3: μ selection to overcome the effect of P and m. As we see in Fig. 3(a), the large P and/or m have the effect of decreasing convergence rate and increasing steady-state MSE. This experiment demonstrates the selection of μ to overcome P and/or m effects. It may be noted that lowering μ by a factor of p (i.e., $1/pN$) would reduce the MSE and misadjustments, but also slows down the convergence rate. We consider p other than power-of-two for PPAT-DLMS, for example, $p = 8/3, 16/3$ by setting $P = 3.9, 4.5$ with $m = 2$, and $P = 3.6, 4.1$ with $m = 3$, as shown in Fig 3(c). Clearly, PPAT-DLMS with $m = 2$ can provide reasonable convergence rate and MSE upto $P = 3.9$ at $\mu = 3/128$. If P is further increased, then μ has to be reduced which could be achieved through the difference of other step-sizes. For instance, μ can be expressed as $\mu = (\frac{1}{2} - \frac{1}{8})/N$ for $p = 8/3$. Clearly, it needs a subtractor after ECU in order to obtain with μ. The similar explanation holds good for $m = 3$, but it possess more misadjustments than $m = 2$.

The existing works [9]–[14] and Structures-I,II and III of [13] are referred as DA_{1-6} and DA_5^{1-3} respectively. $DA_{1,2,6}$ designs have employed large-sized physical LUTs with few adders (ADD), registers (REG) and multiplexers (MUX), as listed in Table I. In $DA_{3,4}$, LUT contents were realized with ADD/REG/MUX, which were then reduced using OBC in DA_5^{1-3}. Note that these designs posses bit-level complexities as denoted by α_k, β_k and γ_k, whose values are listed in footnote of Table III [13]. In DA_{3-5}, several carry-save based SA units were employed which then need an extra adder tree. Compared with DA_5^3, PPAT-DLMS ADF requires less ADD/REG due to a single SA unit for multiple DA bases. For instance, the PPAT-DLMS ADF for $N, L, m = 16, 4, 2$ requires 21.29% lesser ADD and 13.74% lesser REG than DA_5^3 without MUX, as illustrated in Fig. 4. Note that these plots also have normalized bit-level complexities. Similar to DA_5^{1-3}, the proposed design has $(m+1)$ CPs, for instance, CP_{07}, CP_{17} and CP_{27} for $m = 2$, as shown in Fig. 1.

All the designs were coded in Verilog and synthesized using TSMC 90 nm CMOS Library. For fair comparison, LUTs of $DA_{1,2,6}$ were implemented with REG/MUX, while DA_{3-5} were realized as reported at their gate-levels. The ATs of each design was implemented with '+' Verilog operator to maximize the optimization in area, power and minimum sampling period (MSP) through ASIC tools. Note DA_{3-5} were reported for $m = 2$, therefore the location of adaptation delays were also considered same for $m = 4$ except $2\times$ registers for the synthesis purpose. We used the worst case corner (Voltage = 0.9 V, Temperature = 125°C) in all the simulations. In

TABLE II
ASIC SYNTHESIS RESULTS OF DIFFERENT DA BASED ADF ARCHITECTURES USING TSMC 90 NM LIBRARY

N, L, m, μ	$16, 4, 2, 1/16$ (Case 1)					$32, 8, 4, 1/64$ (Case 2)					$64, 8, 4, 3/512$ (Case 3)				
Design	**Area** (μm^2)	**Power** (mW)	**MSP** (ns)	**MSE** (dB)	**EPA** (dB/μm^2)	**Area** (μm^2)	**Power** (mW)	**MSP** (ns)	**MSE** (dB)	**EPA** (dB/μm^2)	**Area** (μm^2)	**Power** (mW)	**MSP** (ns)	MSE (dB)	**EPA** (dB/μm^2)
DA_1 [9]	41053	24.15	15.36	−32.51	-0.79×10^{-3}	118544	73.12	20.98	−58.24	-0.49×10^{-3}	229854	137.56	28.01	−57.67	-0.25×10^{-3}
DA_2 [10]	31547	12.96	28.05	−28.32	-0.89×10^{-3}	89567	36.81	35.26	−51.23	-0.57×10^{-3}	175246	68.42	44.56	−53.88	-0.30×10^{-3}
DA_3 [11]	36078	13.81	4.92	−25.89	-0.71×10^{-3}	103216	39.62	7.97	−48.23	-0.46×10^{-3}	194121	75.11	12.15	−53.29	-0.27×10^{-3}
DA_4 [12]	18269	7.53	10.43	−27.34	-1.49×10^{-3}	59295	19.59	12.11	−50.67	-0.85×10^{-3}	113878	36.52	15.67	−54.65	-0.47×10^{-3}
DA_5^1 [13]	11921	4.59	3.86	−22.67	-1.91×10^{-3}	23812	10.24	5.62	−39.52	-1.65×10^{-3}	44167	18.03	7.93	−48.06	-1.08×10^{-3}
DA_5^2 [13]	10751	4.48	10.48	−23.47	-2.18×10^{-3}	21957	9.81	17.02	−43.24	-1.96×10^{-3}	39769	17.58	26.05	−51.13	-1.28×10^{-3}
DA_5^3 [13]	10492	4.32	9.88	−23.47	-2.23×10^{-3}	20818	9.17	13.04	−43.24	-2.07×10^{-3}	38107	16.34	18.31	−51.13	-1.34×10^{-3}
DA_6 [14]	27872	10.72	29.56	−28.32	-1.01×10^{-3}	80395	30.92	34.89	−51.23	-0.63×10^{-3}	155123	54.81	39.45	−53.88	-0.34×10^{-3}
‡Proposed	8537	3.44	11.78	−31.12	-3.64×10^{-3}	16712	6.83	14.89	−55.47	-3.31×10^{-3}	31847	12.14	19.23	−55.93	-1.75×10^{-3}

this process corner, static power is much lower than dynamic power, and roughly accounts for less than 3% of dynamic power consumption. First, the RTL level Verilog netlist was converted to structural level netlist using the Cadence RTL Compiler tool. The gate level netlist was fully flattened to the basic logic gates. The output Verilog file from the RTL Compiler is then placed and routed using the Cadence Design Encounter tool. The test vectors from the channel output and switching activity files from the post place-and-route (PnR) simulations with full timing and parasitic parameters were obtained. For more accurate estimation, low-driving strength standard cells were used to avoid routing congestion during the PnR. To avoid the timing violations (setup and/or hold), extra buffers were introduced at slight increase in area. The estimated values of area, power MSP and MSE results at 100 MHz system clock after cleaning the DRC/LVS violations from GDSII are listed in Table II. We have considered three cases (1,2,3), i.e., $N, L, m, \mu = (16, 4, 2, 1/16)$, $(32, 8, 4, 1/64)$ and $(64, 8, 4, 3/512)$ respectively with $B/W = 8$ and $B_i/W_i = 0$ at $P = 2.6$. Note that all have different μ $(= 1/pN)$ due to different choice of p, i.e., 1, 2 and 8/3. Based on Table II, the following observations are made:

- Proposed design offers low-area, low-power and low-MSE for all the cases. Compared with DA_5^3, it offers 19.72% less area, 25.51% less power and 28.29% less MSE for case 2. Note that its MSE gain is slightly reduced for larger μ (faster convergence), i.e., with $p = 8/3$.
- Effect of large MSP for proposed design with increasing N can be compensated with smaller μ (or low convergence rate). For instance, the increase in MSP is 26.04% from case 1 to case 2, while it is 63.24% from case 1 to case 3; with effect of L is marginal due to serial LUTs.
- For fair comparison, we define MSE per unit area (EPA) as a figure of merit. It is calculated as:

$$\text{EPA}(\text{dB}/\mu\text{m}^2) = \text{MSE}/\text{Area} \tag{14}$$

Clearly, the proposed design offers 59.91% less EPA over DA_5^3 for the case 2.

VI. CONCLUSION

In this brief, a new DLMS ADF based on TC-DA for fast-convergence and low-MSE has been presented. Parallel predictive adder tree followed by a SA unit with efficient quantization scheme have been used to estimate the response against adaptation delays. Extensive simulations in time-varying band-limited channel, and ASIC synthesis have been carried out to evaluate the performance. It is found that the proposed solution is more attractive for the practical applications.

REFERENCES

[1] S. Haykin, *Adaptive Filter Theory*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 1991.

[2] Y. Kabalci, "5G mobile communication systems: Fundamentals, challenges, and key technologies," in *Smart Grids and Their Communication Systems*. Singapore: Springer, 2019, pp. 329–359.

[3] K. K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*. New Delhi, India: Wiley, 2007.

[4] G. Long, F. Ling, and J. G. Proakis, "The LMS algorithm with delayed coefficient adaptation," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 9, pp. 1397–1405, Sep. 1989.

[5] C. Caraiscos and B. Liu, "A roundoff error analysis of the LMS adaptive algorithm," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 1, pp. 34–41, Feb. 1984.

[6] K. Kammeyer, "Quantization error analysis of the distributed arithmetic," *IEEE Trans. Circuits Syst.*, vol. 24, no. 12, pp. 681–689, Dec. 1977.

[7] S. A. White, "Applications of distributed arithmetic to digital signal processing: A tutorial review," *IEEE ASSP Mag.*, vol. 6, no. 3, pp. 4–19, Jul. 1989.

[8] D. J. Allred, H. Yoo, V. Krishnan, W. Huang, and D. V. Anderson, "LMS adaptive filters using distributed arithmetic for high throughput," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 7, pp. 1327–1337, Jul. 2005.

[9] R. Guo and L. S. DeBrunner, "Two high-performance adaptive filter implementation schemes using distributed arithmetic," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 9, pp. 600–604, Sep. 2011.

[10] M. S. Prakash and R. A. Shaik, "Low-area and high-throughput architecture for an adaptive filter using distributed arithmetic," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 60, no. 11, pp. 781–785, Nov. 2013.

[11] S. Y. Park and P. K. Meher, "Low-power, high-throughput, and low-area adaptive FIR filter based on distributed arithmetic," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 60, no. 6, pp. 346–350, Jun. 2013.

[12] M. T. Khan, R. A. Shaik, and S. P. Matcha, "Improved convergent distributed arithmetic based low complexity pipelined least-mean-square filter," *IET Circuits Devices Syst.*, vol. 12, no. 6, pp. 792–801, 2018.

[13] M. T. Khan and R. A. Shaik, "Optimal complexity architectures for pipelined distributed arithmetic-based LMS adaptive filter," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 2, pp. 630–642, Feb. 2019.

[14] S. Ahmad, S. G. Khawaja, N. Amjad, and M. Usman, "A novel multiplier-less LMS adaptive filter design based on offset binary coded distributed arithmetic," *IEEE Access*, vol. 9, pp. 78138–78152, 2021.

[15] S. C. Douglas and M. Rupp, "Convergence issues in the LMS adaptive filter," in *The Digital Signal Processing Handbook*. Boca Raton, FL, USA: CRC Press, 2017, p. 19.

[16] K. Takahashi, Y. Tsunekawa, N. Tayama, and K. Seki, "Analysis of the convergence condition of LMS adaptive digital filter using distributed arithmetic," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. 85, no. 6, pp. 1249–1256, 2002.

[17] V. H. Nascimento and R. C. de Lamare, "A low-complexity strategy for speeding up the convergence of convex combinations of adaptive filters," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2012, pp. 3553–3556.

[18] G. Di Meo, D. De Caro, G. Saggese, E. Napoli, N. Petra, and A. G. M. Strollo, "A novel module-sign low-power implementation for the DLMS adaptive filter with low steady-state error," *IEEE Trans. Circuits Syst. I, Reg. Papers*, early access, Jun. 24, 2021, doi: 10.1109/TCSI.2021.3088913.

[19] P. Student, "FPGA implementation of efficient VLSI architecture of DLMS adaptive filter algorithm," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 14, pp. 478–489, 2021.

[20] G. Akkad, A. Mansour, B. A. ElHassan, E. Inaty, R. Ayoubi, and J. A. Srar, "A pipelined reduced complexity two-stages parallel LMS structure for adaptive beamforming," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 12, pp. 5079–5091, Dec. 2020.

[21] L.-D. Van and W.-S. Feng, "An efficient systolic architecture for the DLMS adaptive filter and its applications," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 48, no. 4, pp. 359–366, Apr. 2001.