

Very High-Speed VLSI Architecture of Block LMS Adaptive Digital Filter Using Distributed Arithmetic

Kyo Takahashi¹, Yoshitaka Tsunekawa², and Norio Tayama²

¹ Iwate Industrial Technology Junior College, Shiwa, Japan

Tel. +81-19-697-9082, Fax. +81-19-697-9089, E-mail : kyo@iwate-it.ac.jp

²Department of Electrical and Electronic Engineering, Faculty of Engineering, Iwate University, Morioka, Japan

Tel. and Fax. +81-19-621-6468, E-mail: tsune@iwate-u.ac.jp

Abstract: In this paper, we propose a block LMS algorithm using distributed arithmetic (BDA) and a multi-memory block structured BDA (MBDA). Moreover, we propose an effective VLSI architecture of adaptive digital filter using MBDA, and evaluate the sampling rate and output latency.

1. Introduction

In recent years, adaptive digital filters (ADFs) are expected to play an important role in the processing of wide-band signals. In this application, ADFs are especially required to process the signals at very high-speed. We have proposed the LMS adaptive digital filter using distributed arithmetic (DA-ADF) [1], [2]. Our proposed DA-ADF is a high-performance adaptive digital filter that has performances of high speed and small output latency, good convergence speed, small-scale hardware and lower power dissipation for higher order, simultaneously. However, the sampling rate is a few MHz at maximum, so that the DA-ADF is not suitable for higher sampling applications. Block LMS algorithm that is block implementations of the LMS algorithm (BLMS) have been proposed [3]. The block implementations of ADFs allow efficient use of parallel structure, which can result in speed gain.

In this paper, we propose a block LMS algorithm using distributed arithmetic (BDA) and a multi-memory block structured BDA (MBDA) [4]. To enable the pipelined processing, we applied a new update method to these algorithms. We called this method "priority update". Moreover, we propose an efficient VLSI architecture of MBDA-ADF, and evaluate the sampling rate and output latency. As a result, our MBDA-ADF can achieve very high sampling rate and small output latency.

2. Block LMS Algorithm

The general LMS algorithm updates tap coefficients for each input sample [5]. On the other hand, the BLMS with block length L updates tap coefficients every L input samples, so that the tap coefficients are treated as constant during L sample periods. The BLMS with p -tap coefficients is represented as follows [3]. The parameters used in the algorithm are

- $\mathbf{y}_j = [y_{j,0}, \dots, y_{j,(-L+1)}]^T$: output signal vector
- $\mathbf{w}_j = [w_j(0), \dots, w_j(-p+1)]^T$: coefficient vector
- $\mathbf{d}_j = [d_{j,0}, \dots, d_{j,(-L+1)}]^T$: desired signal vector

- $\mathbf{e}_j = [e_{j,0}, \dots, e_{j,(-L+1)}]^T$: error signal vector
 - $\boldsymbol{\varphi}_{j,i} = [x_{j,i}, \dots, x_{j,(i-p+1)}]^T$: input signal vector
 - $\boldsymbol{\Gamma}_j = [\boldsymbol{\varphi}_{j,0}, \dots, \boldsymbol{\varphi}_{j,(-L+1)}]^T$: input signal matrix
- , where j , p and L indicate block number, tap number and block length, respectively. The sampling time k of input signal $x_{j,i}$ is

$$k = jL + i, \quad i = 0, -1, \dots, L-1. \quad (1)$$

The L -outputs \mathbf{y}_j and L -errors \mathbf{e}_j for the time from $(j-1)L+1$ to jL is obtained as

$$\mathbf{y}_j = \boldsymbol{\Gamma}_j \mathbf{w}_j, \quad (2)$$

$$\mathbf{e}_j = \mathbf{d}_j - \mathbf{y}_j, \quad (3)$$

and the BLMS is represented as

$$\mathbf{w}_{(j+1)} = \mathbf{w}_j + \frac{2\mu_B}{L} \boldsymbol{\Gamma}_j^T \mathbf{e}_j. \quad (4)$$

3. BDA Algorithm

The distributed arithmetic (DA) is well-known as an efficient calculation method of an inner product for not only constant vector but time varying coefficient vector [6], [1], [7]. In the DA, the inner product is obtained by the shift and accumulation of partial products for the word length. For the tap number p , there exist 2^p partial products corresponding to the bit variations of p -th order vector. The set of 2^p partial products is called Whole Adaptive Function Space (WAFS). WAFS is previously determinable for the constant vector. However, for the time varying tap coefficients, WAFS is estimated using the adaptive algorithm.

The BDA is derived by applying the DA to the BLMS. We indicate the input signal $\boldsymbol{\varphi}_{j,i}$ as

$$\boldsymbol{\varphi}_{j,i} = \mathbf{A}_{j,i} \mathbf{F}, \quad i = 0, -1, \dots, -L+1, \quad (5)$$

where, the address matrix $\mathbf{A}_{j,i}$ and scaling vector \mathbf{F} are

$$\mathbf{A}_{j,i} = \begin{bmatrix} b_{j,i}(0) & \dots & b_{j,(i-p+1)}(0) \\ b_{j,i}(1) & \dots & b_{j,(i-p+1)}(1) \\ \vdots & \ddots & \vdots \\ b_{j,i}(B-1) & \dots & b_{j,(i-p+1)}(B-1) \end{bmatrix}^T, \quad (6)$$

$$\mathbf{F} = [-2^0, 2^{-1}, \dots, 2^{-(B-1)}]^T, \quad (7)$$

where B and $b_{j,i}(l)$ indicate word length and l -th bit of the input signal $x_{j,i}$. An address vector (AV) is defined as the column vector of the address matrix, i.e.,

$$\mathbf{Av}_{j,i}(l) = [b_{j,i}(l), b_{j,i-1}(l), \dots, b_{j,i-p+1}(l)]^T, \quad l = 0, 1, \dots, B-1, \quad (8)$$

and the value of AV is defined as

$$\mathbf{Av}_{j,i}(l) = \mathbf{Av}_{j,i}^T(l) \mathbf{F}_A, \quad (9)$$

$$\mathbf{F}_A = [2^{(p-1)}, 2^{(p-2)}, \dots, 2^0]^T. \quad (10)$$

The partial products are defined for the AV, so that the AV is used to specify the partial product of WAFS.

From above discussion, the BLMS is represented as

$$\mathbf{w}_{(j+1)} = \mathbf{w}_j + \frac{2\mu_B}{L} \sum_{i=0}^{-L+1} \mathbf{A}_{j,i} \mathbf{F} e_{j,i}. \quad (11)$$

For i , this equation is expanded to L equations, i.e.,

$$\begin{aligned} \mathbf{w}_{j,(-L+2)} &= \mathbf{w}_{j,(-L+1)} \\ &+ \frac{2\mu_B}{L} \mathbf{A}_{j,(-L+1)} \mathbf{F} e_{j,(-L+1)}, \quad (12) \end{aligned}$$

$$\begin{aligned} &\vdots \\ \mathbf{w}_{j,0} &= \mathbf{w}_{j,(-1)} + \frac{2\mu_B}{L} \mathbf{A}_{j,(-1)} \mathbf{F} e_{j,(-1)}, \quad (13) \end{aligned}$$

$$\mathbf{w}_{(j+1),(-L+1)} = \mathbf{w}_{j,0} + \frac{2\mu_B}{L} \mathbf{A}_{j,0} \mathbf{F} e_{j,0}, \quad (14)$$

where $\mathbf{w}_{j,i}$ indicates the tap coefficient vector at the time i in the block j , and has relationship

$$\mathbf{w}_{(j+1)} = \mathbf{w}_{(j+1),(-L+1)}. \quad (15)$$

Although the DA is applied from equation(12) to equation(14), here, we show the derivation only for equation(12). Multiplying the both sides by $\mathbf{A}_{j,i}$ from the left, equation(12) becomes

$$\begin{aligned} \mathbf{A}_{j,(-L+1)}^T \mathbf{w}_{j,(-L+2)} &= \mathbf{A}_{j,(-L+1)}^T \mathbf{w}_{j,(-L+1)} \\ &+ \frac{2\mu_B}{L} \mathbf{A}_{j,(-L+1)}^T \mathbf{A}_{j,(-L+1)} \mathbf{F} e_{j,(-L+1)}. \quad (16) \end{aligned}$$

We define an adaptive function space (AFS) that is a subset of the WAFS selected by $\mathbf{Av}_{j,i}(l)$ as follows.

$$\begin{aligned} \mathbf{P}_{j,i}^n &\equiv \mathbf{A}_{j,n}^T \mathbf{w}_{j,i} \\ &= [p_{j,i}(Av_{j,i}(0)), \dots, p_{j,i}(Av_{j,i}(B-1))]^T \quad (17) \end{aligned}$$

$$\begin{aligned} \mathbf{P}_{j,(i+1)}^n &\equiv \mathbf{A}_{j,n}^T \mathbf{w}_{j,(i+1)} \\ &= [p_{j,(i+1)}(Av_{j,i}(0)), \dots, p_{j,(i+1)}(Av_{j,i}(B-1))]^T \quad (18) \end{aligned}$$

Applying the above definitions, equation(12) becomes

$$\begin{aligned} \mathbf{P}_{j,(-L+2)}^{(-L+1)} &= \mathbf{P}_{j,(-L+1)}^{(-L+1)} \\ &+ \frac{2\mu_B}{L} \mathbf{A}_{j,(-L+1)}^T \mathbf{A}_{j,(-L+1)} \mathbf{F} e_{j,(-L+1)}. \quad (19) \end{aligned}$$

Moreover, the following relation [1]

$$E[\mathbf{A}_{j,i}^T \mathbf{A}_{j,i}] = 0.25p \mathbf{F} \quad (20)$$

is applied to equation(12).

Applying the same procedure to equation(13) and equation(14), we obtain the BDA as follows.

$$\mathbf{P}_{j,(-L+2)}^{(-L+1)} = \mathbf{P}_{j,(-L+1)}^{(-L+1)} + \mathbf{u}_{j,(-L+1)}, \quad (21)$$

$$\begin{aligned} &\vdots \\ \mathbf{P}_{j,0}^{(-1)} &= \mathbf{P}_{j,(-1)}^{(-1)} + \mathbf{u}_{j,(-1)}, \quad (22) \end{aligned}$$

$$\mathbf{P}_{(j+1),(-L+1)}^0 = \mathbf{P}_{j,0}^0 + \mathbf{u}_{j,0}, \quad (23)$$

where

$$\begin{aligned} \mathbf{u}_{j,i} &= 0.5p \frac{\mu_B}{L} \mathbf{F} e_{j,i} \\ &= [u_{j,i}(0), u_{j,i}(1), \dots, u_{j,i}(B-1)]^T. \quad (24) \end{aligned}$$

The output equation is

$$\begin{aligned} \mathbf{y}_j &= [y_{j,0}, y_{j,(-1)}, \dots, y_{j,(-L+1)}]^T \\ &= [\mathbf{F}^T \mathbf{P}_{j,0}^0, \mathbf{F}^T \mathbf{P}_{j,(-1)}^{(-1)}, \dots, \mathbf{F}^T \mathbf{P}_{j,(-L+1)}^{(-L+1)}]^T. \quad (25) \end{aligned}$$

The BDA is not suited for parallel and pipeline processing [4]. To overcome this problem, the BDA is extended to the equations for WAFS. Besides, we proposed a new update method that the update value for one partial product is generated by only the largest scaled error in each update equation. We call this "priority update method" [4]. The BDA using the priority update represented as

$$\mathbf{P} \mathbf{w}_{j,(-L+2)} = \mathbf{P} \mathbf{w}_{j,(-L+1)} + \mathbf{U}_{j,(-L+1)}, \quad (26)$$

$$\begin{aligned} &\vdots \\ \mathbf{P} \mathbf{w}_{j,0} &= \mathbf{P} \mathbf{w}_{j,(-1)} + \mathbf{U}_{j,(-1)}, \quad (27) \end{aligned}$$

$$\mathbf{P} \mathbf{w}_{(j+1),(-L+1)} = \mathbf{P} \mathbf{w}_{j,0} + \mathbf{U}_{j,0}, \quad (28)$$

where $\mathbf{P} \mathbf{w}_{j,i}$ indicates the WAFS as

$$\mathbf{P} \mathbf{w}_{j,i} = [p_{j,i}(0), p_{j,i}(1), \dots, p_{j,i}(2^p - 1)]^T \quad (29)$$

and

$$\begin{aligned} \mathbf{U}_{j,i} &= [u_{j,i}(0), u_{j,i}(1), \dots, u_{j,i}(2^p - 1)]^T \\ &= \mathbf{T}_{j,i} \{0.5p \frac{\mu_B}{L} \mathbf{F} e_{j,i}\}. \quad (30) \end{aligned}$$

$\mathbf{T}_{j,i}$ is a transfer matrix size of $2^p \times B$ which is determined by the address vectors.

4. MBDA Algorithm

WAFS must be of length 2^p words to accommodate the set of partial products corresponding to the set of all possible address words, so memory space required will become impractically large for higher order. Besides the convergence speed drastically degrade, because the probabilities of the updating partial products become

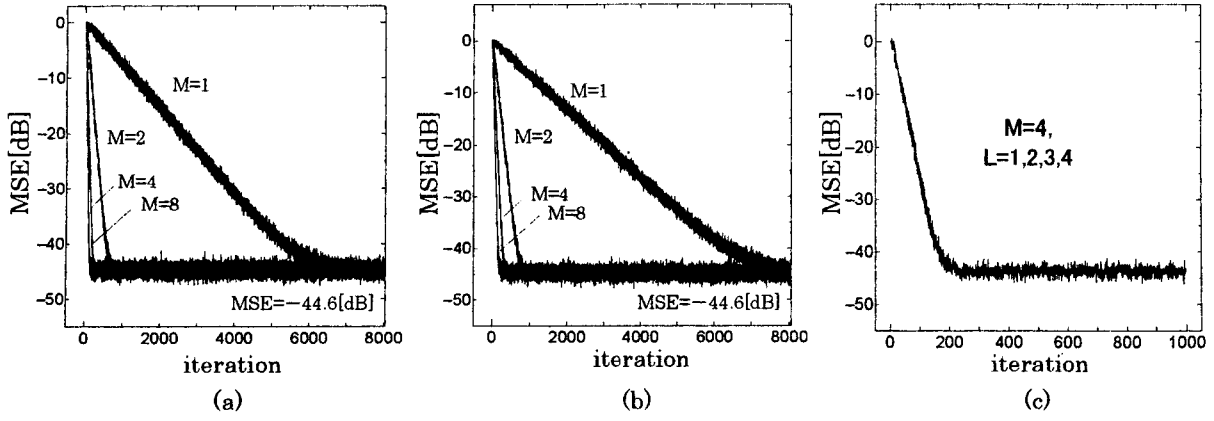


Figure 1. Convergence characteristics. (a) MDA with $M=1,2,4,8$. (b) proposed MBDA with $L=4$ and $M=1,2,4,8$. (c) proposed MBDA with $L=1,2,3,4$ and $M=4$.

smaller. To overcome this problem, the multi-memory block structure, which has M -divided small WAFSSs, has been proposed [7]. The MBDA applied multi-memory block structure to BDA is represented as follows [4]. The divided tap coefficients vector and WAFS are defined as

$$\mathbf{w}_j^m = [w_j^m(0), \dots, w_j^m(2^R - 1)]^T, \quad (31)$$

$$\mathbf{P}w_j^m = [p_j^m(0), \dots, p_j^m(2^R - 1)]^T, \quad (32)$$

where $R = p/M$, $m = 0, 1, \dots, M - 1$. The update equations are

$$\mathbf{P}w_{j,(-L+2)}^m = \mathbf{P}w_{j,(-L+1)}^m + \mathbf{U}_{j,(-L+1)}^m, \quad (33)$$

⋮

$$\mathbf{P}w_{j,0}^m = \mathbf{P}w_{j,(-1)}^m + \mathbf{U}_{j,(-1)}^m, \quad (34)$$

$$\mathbf{P}w_{(j+1),(-L+1)}^m = \mathbf{P}w_{j,0}^m + \mathbf{U}_{j,0}^m, \quad (35)$$

where

$$\begin{aligned} \mathbf{U}_{j,i}^m &= \mathbf{T}_{j,i}^m [0.5R \frac{\mu B}{L} \mathbf{F} e_{j,i}] \\ &= [u_{j,i}^m(0), u_{j,i}^m(1), \dots, u_{j,i}^m(2^R - 1)]^T. \end{aligned} \quad (36)$$

$\mathbf{T}_{j,i}^m$ is a transfer matrix size of $2^R \times B$ determined by the address vectors.

5. Convergence Properties

We simulate the system identification problem. The unknown system is a low-pass FIR filter with 8-taps, and the input signal is a white gaussian noise with zero-mean. And the observation noise is a white gaussian noise independent to the input signal with zero-mean and variance of 1.50998×10^{-6} . Figure. 1 (a) shows the convergence characteristics of Multi-memory block structured DA-ADF [1] with $M = 1, 2, 4, 8$, where, the characteristics for $M = 8$ is equivalent to the BLMS. Figure. 1 (b) shows characteristics of proposed MBDA with $M = 1, 2, 4, 8$ and $L = 4$. From these figures, we can see that the MBDA has good performances nearly equivalent to MDA algorithm. Moreover, Figure. 1 (c) shows the convergence characteristics for $M = 4$ and $L = 1, 2, 3, 4$. From this figure, we

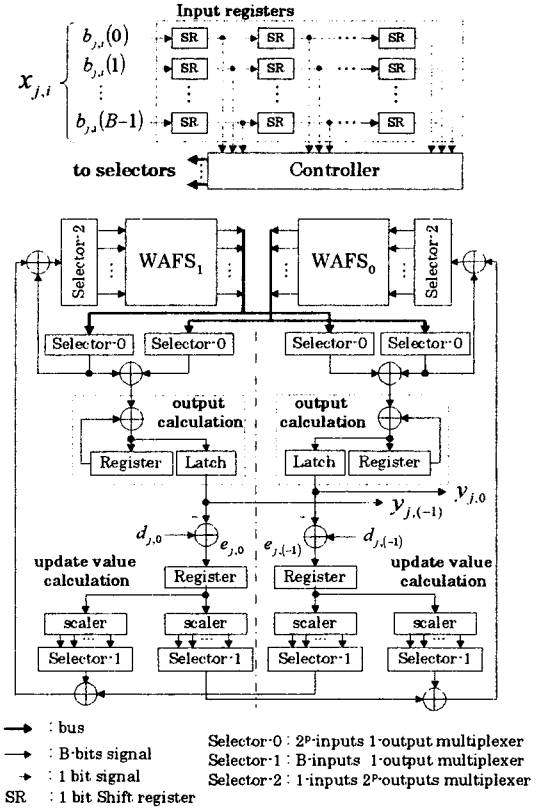


Figure 2. Proposed architecture of MBDA-ADF with $L=M=2$.

can see that the MBDA also has good characteristics for various block lengths.

We obtained the same results for many cases in computer simulations.

6. VLSI Architecture and Evaluations

We show the proposed VLSI architectures in Fig. 2, where $L = M = 2$. The unit of input registers consists of $(p+L-1) \times B$ 1-bit shift registers (SR). The controller generates the select signals for the selector-0, selector-1 and selector-2. There are 2 WAFSSs for $M = 2$, and 2 sets of output calculation and update units for $L = 2$

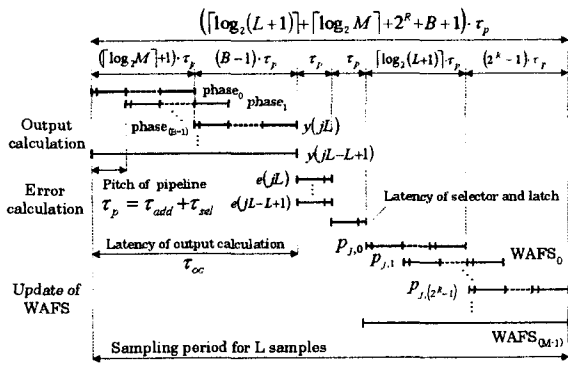


Figure 3. Timing chart of proposed MBDA-ADF.

been placed on both sides of the dashed line.

The MBDA-ADF performs 2 operations as follows.

[Output calculation]

The following procedures are executed in B times.

- Selection of the elements of WAFS₀ and WAFS₁ using the selector-0s.
- Addition of the selected 2 elements.
- Shift and accumulation of the sum.

[Update]

The following procedures are executed for the elements from 0 to 2^{R-1} -th of the WAFS₀ and WAFS₁.

- Selection of the update values from scaler outputs to update i -th element using the selector-1s.
- Addition of the 2 selected update values corresponding to m .
- Summation of the update values obtained above and the i -th element of WAFS, and storing.

Figure. 3 shows the timing chart of MBDA-ADF. From this figure, the sampling period T_s , sampling rate F_s and output latency τ_o per 1 input sample are

$$T_s = ([\log_2(L+1)] + [\log_2 M] + 2^R + B + 1) \cdot \tau_p / L, \quad (37)$$

$$F_s = 1/T_s, \quad (38)$$

$$\begin{aligned} \tau_o &= L \times T_s + \tau_{oc} \\ &= ([\log_2(L+1)] + 2[\log_2 M] + 2^R + 2B + 1) \cdot \tau_p. \end{aligned} \quad (39)$$

TABLE 1 shows the comparison of the sampling rate and output latency between BLMS-ADF [3] and MBDA-ADF, where $p = 128$, $\tau_{add} = 15$ ns and $\tau_{sel} = 7$ ns. We selected p and L to the same value of positive integer power of 2, because Clark and others applied BLMS in the frequency domain, which used Fast Fourier Transform, to ADF. From this Table, MBDA-ADF can achieve very high sampling rate of 165.5MHz (277.7% of BLMS) and small output latency of 1259.7ns (39.5% of BLMS). For larger L , MBDA-ADF can achieve higher sampling rate.

7. Conclusions

In this paper, we have proposed the new block LMS algorithms using distributed arithmetic, BDA and MBDA. Our MBDA is suitable for pipeline processing,

Table 1. Comparison of sampling rate F_s and output latency τ_o between MBDA-ADF with $M = 64$ and BLMS-ADF. The word length $B = 16$.

L(=p)	MBDA		BLMS	
	Fs [MHz]	τ_o [ns]	Fs [MHz]	τ_o [ns]
8	13.4	994.5	7.04	1674.0
16	25.0	1060.8	11.5	2052.0
32	46.7	1127.1	19.5	2430.0
64	87.8	1193.4	33.8	2808.0
128	165.5	1259.7	59.6	3186.0

and has good convergence characteristics. Moreover, we have proposed an effective VLSI architecture using MBDA. We have confirmed that the MBDA-ADF can achieve very high sampling rate of 165.5MHz (277.7% of BLMS) and small output latency of 1259.7ns (39.5% of BLMS) from our evaluations.

Considerations on the detailed VLSI evaluations and convergence condition are considered as future works.

References

- [1] Y. Tsunekawa, K. Takahashi, S. Toyoda, M. Miura, "High-Performance VLSI Architecture of Multiplierless LMS Adaptive Filters Using Distributed Arithmetic," IEICE Trans. Fundamentals, vol. J82-A, no. 10, pp.1518-1528, Oct. 1999.
- [2] K. Takahashi, Y. Tsunekawa, N. Tayama, K. Seki, "Analysis of the Convergence Condition of LMS Adaptive Filter Using Distributed Arithmetic," IEICE TRANS. FUNDAMENTALS, vol. E85-A, NO. 6, pp. 151-158, JUNE. 2002.
- [3] Gregory A. Clark, Sanjit K. Mitra, Sydney R. Parker, "Block Implementation of Adaptive Digital Filters," IEEE Trans. Circuits and Syst., vol. CAS-28, pp.584-592, June. 1981.
- [4] K. Takahashi, N. Higuchi, Y. Tsunekawa, N. Tayama, "Very High-Speed VLSI Architecture of Block LMS Adaptive Filter Using Distributed Arithmetic," Proceedings of the 201-th SICE Tohoku-Branch Research Convention, vol. 201-6, May. 2002.
- [5] B. Widrow, J. R. Glover, Jr., J. M. McCool, J. Kautitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, E. Dong, Jr., and R. C. Goodlin, "Adaptive noise cancelling: Principles and applications," Proc. IEEE, vol. 63, pp.1692-1716, Dec. 1975.
- [6] A. Peled and B. Liu, "A new hardware realization of digital filters," IEEE Trans. Acoust., Speech & Signal Process., vol. 22, no. 12, pp. 456-462, Dec. 1974.
- [7] C. F. N. Cowan and J. Mavor, "New digital adaptive filter implementation using distributed-arithmetic techniques," IEE Proc., vol. 128, Pt. F, no. 4, pp. 225-230, Aug. 1981.
- [8] C. H. Wei, J. J. Lou, "Multimemory block structure for implementing a digital adaptive filter using distributed arithmetic," IEE Proc., vol. 133, Pt. G, no. 1, pp. 19-26, Feb. 1986.