Fast Convergent Pipelined Adaptive DFE Architecture Using Post-Cursor Processing Filter Technique

Meng-Da Yang, An-Yeu (Andy) Wu, and Jyh-Ting (Justin) Lai

***Abstract*—Among existing works of high-speed pipelined adaptive decision feedback equalizers (ADFE), the pipelined ADFE using relaxed look-ahead technique results in a substantial hardware saving than the parallel processing or other look-ahead approaches. For example, Shanbhag and Partiz derived three pipeline ADFE structures (PIPEADFE1, 2, 3), where PIPEADFE2 yields very good performance in terms of convergence rate and hardware cost. Nevertheless, the PIPEADFE2 employs *Approximation Methods* in deriving the Preprocessing Unit (PP). In this paper, a new pipelining ADFE architecture is developed. We derive a new updating ADFE scheme based on the *Principle of Orthogonality*. By employing the postcursor processing filter (PCF) to cancel the most significant postcursor Intersymbol interference (ISI) terms, the proposed PCFADFE architecture can significantly improve the convergence rate of the ADFE. Compared with PIPEADFE2, it has better convergence rate while at similar hardware cost. Hence, it provides an alternative approach for the design of high-speed pipelining ADFE with arbitrary speedup factor.**

***Index Terms*—Decision feedback equalizer, postcursur processing filter.**

I. Introduction

ADAPTIVE *decision feedback equalizer (ADFE)* using *least mean-squared (LMS)* algorithm is a popular equalization technique for magnetic storage and digital communication systems [1]–[3]. However, the fine-grain pipelining of the ADFE is known to be a difficult problem for high-speed applications. According to the *iteration bound* [4], the operating clock rate of the ADFE is limited by the *decision feedback loop (DFL)*. Several approaches have been proposed to solve aforementioned problems. For example, pipelining the ADFE can be achieved by precomputing all possibilities in DFL to open the DFL [5]. However, it results in a significant hardware overhead as it transforms a serial algorithm into an equivalent (in the sense of *input–output behavior*) pipelined algorithm. Three new adaptive pipelined DFE designs (PIPEADFE1,2,3) were proposed in [6]. They maintain the functionality in the *statistical behavior* instead of *input–output behavior* by using the *relaxed look-ahead* technique. From VLSI implementation point of view, the PIPEADFE2 is more suitable for the low-cost VLSI implementations with good signal-to-noise ratio (SNR) and convergence prosperities. Nevertheless, the PIPEADFE2 is derived based on the *Approximation Methods* under some pre-assumed statistical models.

In this paper, we propose an alternative ADFE architecture to improve the convergence rate of the Relaxed Look-ahead ADFE. Our derivation is based on the *Principle of Orthogonality* rather than *Approximation Methods*. Hence, it provides more insightful information for the derived ADFE algorithms and architectures. From the derivation, we introduce the *postcursor processing filter (PCF)* technique to cancel the most significant postcursor intersymbol interference (ISI) terms. The PCF-based AFFE (PCFADFE) is similar to the PIPEADFE2 of [6] but with more degrees of freedom in updating the ADFE; therefore, it leads to better convergence properties. The simulation results show that the proposed PCFADFE architecture can converge faster than PIPEADFE2 of [6] at the same SNR requirement, whereas the hardware complexities of both architectures are close. Hence, the proposed scheme provides an alternative way to design very high-speed ADFE with arbitrary speed-up factor.

II. Review of Pipelined ADFE Architectures

In [6], the *delayed LMS* [8] and the technique of *transfer delay relaxation* [9] are employed to develop the PIPEADFE1 and PIPEADFE2. In addition, the technique of *sum relaxation* [6] is applied to pipeline the updating circuit of ADFE. The resulting pipelined ADFE architecture PIPEADFE1 and PIPEADFE2 are shown in Figs. 1 and 2, respectively. Basically, the PIPEADFE1 is an extension of the ADFE algorithm proposed in [7]. PIPEADFE2 is a variant of PIPEADFE1. It employs a *Preprocessing Unit (PP)* before FFE to alleviate the burden of the FFE. Note that the PP section copies the first D_1 coefficients of the FBE, hence, PP does not need extra weight update circuit. However, PIPEADFE2 loses degree of freedom in the weight space.

III. Proposed Pipelined Adaptive DFE Architecture

In [6], it has been shown that the convergence rate of PIPEADFE1 is quite slower than the serial ADFE. In the case of PIPEADFE1, the FBF cannot cancel the first D_1 postcursor ISI terms and the burden of canceling those postcursor ISI falls on the FFF. The authors in [6] also propose PIPEADFE2 that can handle the problem of PIPEADFE1. However, it is based on the *Approximation Method* under assumed statistical models. Here, we propose an alternate approach. It applies the PCF to the PIPEADFE1 to cancel the first D_1 postcursor ISI terms instead of using the FFF. Hence, the burden of FFF can be alleviated.

The derivation of the proposed algorithm is as follows. Assume that the difference between the slicer input and output is

Manuscript received April 8, 2002; revised September 16, 2003. This work was supported in part by the National Science Council, R.O.C. under Grant 91-2218-E-002-029. This paper was recommended by Associate Editor K. Parhi.

The authors are with the Graduate Institute of Electronics Engineering, and Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan, R.O.C.

Digital Object Identifier 10.1109/TCSII.2003.822421

1057-7130/04$20.00 © 2004 IEEE

Authorized licensed use limited to: National Taiwan University. Downloaded on March 2, 2009 at 02:33 from IEEE Xplore. Restrictions apply.

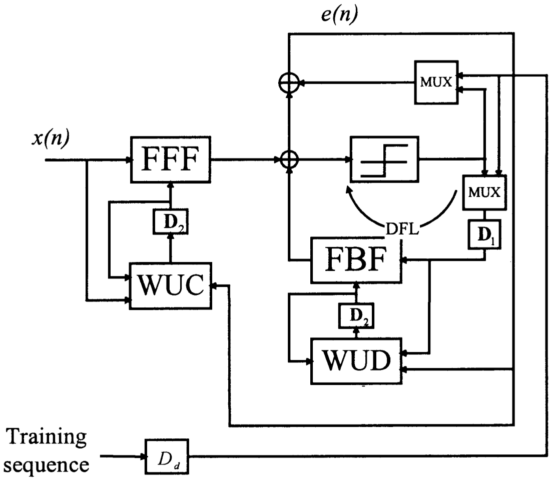

Fig. 1. PIPEADFE1 architecture [6], where FFF and FBE denote the and *feedback filter*, and WUC and WUD are the *weight update circuits* of FFF and FBE, respectively.

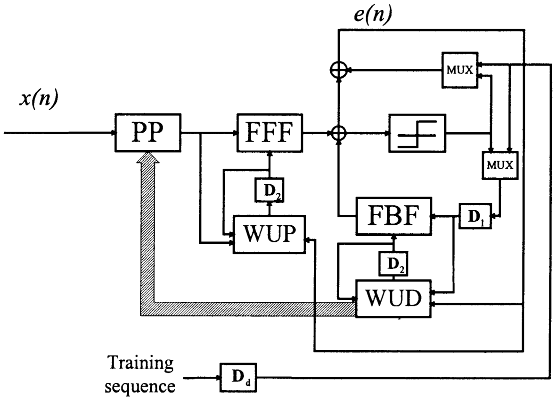

Fig. 2. PIPEADFE2 architecture [6].

small. The behavior of DFE is close to an IIR filter with the transfer function

$$P(z) = \frac{N(z)}{D(z)}. \tag{1}$$

Then, the traditional way of pipelining IIR filter can be applied here [3]. That is, by inserting the same poles and zeros pairs, we can decorrelate the dependence of the output and the first D_1 ISI terms. We can find the polynomial $Q(z) = \sum_{i=0}^{D_1} q_i z^{-i}$, $q_0 = 1$, and multiply it to the denominator and numerator of original IIR transfer function, $P(z)$, to pipeline the DFL. This results in

$$P(z) = \frac{N(z)}{D(z)} = \frac{Q(z)N(z)}{Q(z)D(z)} = \frac{Q(z)N(z)}{1 - z^{-(D_1+1)}R(z)} \tag{2}$$

with $R(z) = \sum_{i=0}^{k} r_i z^{-i}$. By using these extra D_1 delay elements, we can apply the retiming techniques [4]. Then, the highest operating clock rate of DFE can be increased by a factor of $(D_1 + 1)$. In our paper, we call $Q(z)$ *postcursor processing filter (PCF)*. By solving the linear equations, we can obtain the coefficients of $Q(z)$. However, this pipelined DFE algorithm is inherently nonadaptive. In the remainder of this section, we will combine this pipelined DFE algorithm and adaptive filter algorithm to derive PCFADFE.

A. Derivation of PCFADFE With $D_1 = 2$

To illustrate our design, we first show an example of PCFADFE with a speedup factor of three $(D_1 = 2)$. That is, iteration bound of serial ADFE is three times than the PCFADFE. This implies the two extra delay elements $(D_1 = 2)$ must be inserted into the DFL. Assume that the transmitted data, $a(n)$, is an independent sequence, and input data of receiver is $x(n)$. Then we have

$$\begin{aligned} x(n) = h_{-1}a(n+1) + h_0 a(n) \\ + h_1 a(n-1) + h_2 a(n-2) + \nu(n) \end{aligned} \tag{3}$$

where h_{-1}, h_0, h_1, h_2, are channel impulse response, $v(n)$ is the additive Gaussian white noise. The number of taps in FFF and FBF are three and two, respectively. Since $D_1 = 2$, the number of taps in the PCF is three in this example. The i-th coefficients of FFF, PCF and FBF at time instance n are denoted as c_i, p_i, and b_i, respectively. With above notations, the estimation error $e(n)$ can be expressed as

$$\begin{aligned} e(n) &= a(n) - F(n) - P(n) \\ &= a(n) - F(n) - p_1 F(n-1) \\ &\quad -p_2 F(n-2) - B(n) \end{aligned} \tag{4a}$$

$$F(n) = \sum_{i=0}^{2} c_i x(n+i) \tag{4b}$$

$$B(n) = \sum_{i=1}^{2} b_i a(n-2-i) \tag{4c}$$

where $F(n)$ is the output of FFF, $B(n)$ is the output of FBF. $P(n)$ denotes the total effect of PCF and FBF at time instance n, and can be written as

$$\begin{aligned} P(n) &= p_1 F(n-1) + p_2 F(n-2) + B(n) \\ &= \sum_{i=-2}^{4} s_i a(n-1) + \eta(n) \end{aligned} \tag{5}$$

where $\eta(n)$ is the noise constant.

Let us consider the updating equations of the PCF and FBF. From (3), the FFF output at time instance n can be written as

$$F(n) = \sum_{i=0}^{2} c_i x(n+i) = \sum_{i=-3}^{2} r_i a(n-i). \tag{6}$$

$F(n)$ represents sum of the residual ISI terms that cannot be canceled by FFF at time instance n. Because we employ a dedicated PCF to de-correlate the relationship between the first two post-cursor ISI terms and ADFE output, the remaining ISI terms will be canceled by FFF and FBF, respectively. Recall that, in the serial ADFE algorithm, the prediction error must be orthogonal to the observations in the steady state. This is so-called "*Principle of Orthogonality*" in the literature of adaptive signal processing [10]. It implies that minimizing the estimation error is equivalent to de-correlate the relationship between observations and filter output. Therefore, the objective of the PCF is to minimize the following two expectation terms of

$$\mathrm{Min}_{p_1}\{\mathbf{E}^2\{e(n)a(n-1)\}\} \tag{7a}$$

$$\mathrm{Min}_{p_2}\{\mathbf{E}^2\{e(n)a(n-2)\}\} \tag{7b}$$

Authorized licensed use limited to: National Taiwan University. Downloaded on March 2, 2009 at 02:33 from IEEE Xplore. Restrictions apply.

where p_1, p_2 are the coefficients of the 2—tap PCF. Next, the gradients corresponding to these costs functions are

$$\begin{aligned}\frac{\partial\mathbf{E}\{(a(n-1)e(n))^2\}}{\partial p_1} &= -2r_0\mathbf{E}\{e(n)a(n-1)\}\\ &\quad\times\mathbf{E}\{a^2(n-1)\}\\ &=\mu\mathbf{E}\{e(n)a(n-1)\} \qquad (8a)\\ \frac{\partial\mathbf{E}\{(a(n-2)e(n))^2\}}{\partial p_2} &= -2r_0\mathbf{E}\{e(n)a(n-2)\}\\ &\quad\times\mathbf{E}\{a^2(n-2)\}\\ &=\mu\mathbf{E}(e(n)a(n-2)\}. \qquad (8b)\end{aligned}$$

Here we lump $-2r_0\mathbf{E}\{a^2(n-1)\}$ as a constant μ. The reason is that the direction of gradient is more important than the magnitude of gradient in stochastic gradient-based algorithm.

Similarly, in the FBF, we intend to minimize the same cost function of FFF. The gradients corresponding to FBF can be derived as

$$\frac{\partial\mathbf{E}\{e^2(n)\}}{\partial b_1} = -2\mathbf{E}\{e(n)a(n-3)\} \qquad (9a)$$

$$\frac{\partial\mathbf{E}\{e^2(n)\}}{\partial b_2} = -2\mathbf{E}\{e(n)a(n-4)\}. \qquad (9b)$$

B. Generalized PCFADFE Algorithm for an Arbitrary Speedup Factor

The equations we derived for this particular example of $D_1 = 2$ can be generalized to the general case with arbitrary taps and arbitrary speedup factor. Finally, combining with *Sum Relaxed Look-ahead technique* and the generalized cases of (6), (8), and (9), the equations to describe the proposed PCFADFE algorithm can be written as

$$\mathbf{X}(n) = [x(n)\cdots x(n-N_f+1)]^T \qquad (10a)$$

$$\mathbf{Y}(n) = [\hat{a}(n-D_1-1)\cdots\hat{a}(n-D_1-N_b)]^T \qquad (10b)$$

$$\mathbf{Z}(n) = [\,\hat{a}(n-1)\ \cdots\ \cdots\ \hat{a}(n-D_1)\,]^T \qquad (10c)$$

$$\mathbf{P}(n) = [\,p_1(n)\ \cdots\ \cdots\ p_{D_1}(n)\,]^T \qquad (10d)$$

$$F(n) = \mathbf{C^T}(n-D_2)\mathbf{X}(n) \qquad (10e)$$

$$B(n) = \mathbf{D^T}(n-D_2)\mathbf{Y}(n) \qquad (10f)$$

$$\tilde{a}(n) = \sum_{j=0}^{D_1} p_j(n-D_2)F(n-j) + B(n),$$

$$p_0 = 1 \qquad (10g)$$

$$\hat{a}(n) = Q[a(n)] \qquad (10h)$$

$$e(n) = \hat{a}(n) - \tilde{a}(n) \qquad (10i)$$

$$\mathbf{C}(n) = \mathbf{C}(n-D_2) + \mu\sum_{i=0}^{LA-1} e(n-i)\mathbf{X}(n-i) \qquad (10j)$$

$$\mathbf{D}(n) = \mathbf{D}(n-D_2) + \mu\sum_{i=0}^{LA-1} e(n-i)\mathbf{Y}(n-i) \qquad (10k)$$

$$\mathbf{P}(n) = \mathbf{P}(n-D_2) + \left(\frac{\mu}{N}\right)\sum_{i=0}^{LA-1} e(n-i)\mathbf{Z}(n-i). \qquad (10l)$$

Note that the (10l) using a smaller step size to maintain the converged SNR, and p_j denotes the j-th coefficient of the PCF. The corresponding hardware architecture of PCFADFE is shown in Fig. 3.

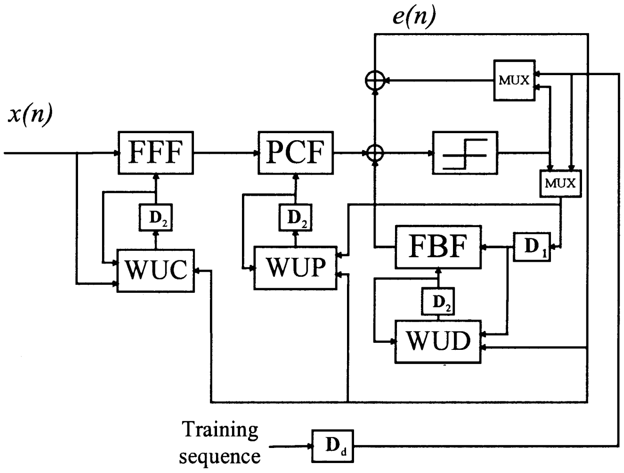

Fig. 3. The architecture of the proposed PCFADFE.

TABLE I
PARAMETERS SETTING FOR CHANNEL I WITH (C) $W = 3.5$, AND (D) CHANNEL II

	N_f	N_b	D_1	Step size	Output SNR	Other parameters
Serial ADFE	13	10	0	$2^{-6.5}$	19.1 dB	$LA=0, D_2=1, D_d=8$
PIPEADFE1	13	10	6	$2^{-6.5}$	15.2 dB	$LA=0, D_2=1, D_d=8$
PIPEADFE2	13	10	6	$2^{-6.5}$	15 dB	$LA=0, D_2=1, D_d=8$
PCFADFE	13	10	6	$2^{-6.5}$	14.8 dB	$LA=0, D_2=1, D_d=8, N=6$

	N_f	N_b	D_1	Step size	Output SNR	Other parameters
Serial ADFE	13	10	0	2^{-7}	16.8 dB	$LA=0, D_2=1, D_d=8$
PIPEADFE1	13	10	6	2^{-7}	14.1 dB	$LA=0, D_2=1, D_d=8$
PIPEADFE2	13	10	6	2^{-7}	14 dB	$LA=0, D_2=1, D_d=8$
PCFADFE	13	10	6	2^{-7}	13.9 dB	$LA=0, D_2=1, D_d=8, N=6$

IV. SIMULATION RESULTS AND COMPARISONS

In this section, we will show that the convergence rate of the proposed PCFADFE is faster than PIPEADFE1 and PIPEADFE2 by simulation results. This improvement comes from the post-cursor filtering scheme in PCFADFE. In our simulation, we will consider two types of channel models. In first channel model (Channel I), we adopt simple low pass channel with eigenvalue spread 3.3 [10, Sec. 9]. The second channel impulse response (Channel II) is the typical channel impulse response of UTP-CAT-5, which is often employed in Ethernet applications. The transmitted data $a(n)$ are a binary random sequence, *i.e.*, $a(n) \in \{-1, 1\}$.

A. Convergence Rate of Serial-ADFE, PIPEADFE1, PIPEADFE2, and PCFADFE

In this simulation, we evaluate the convergence performance of all equalizers with input $\mathrm{SNR} = 22$ dB. The parameter settings for these four channel models are listed in Table I. With the parameters setting, the learning curves of Serial-ADFE, PIPEADFE1, PIPEADFE2, and PCFADFE for the target channel models are shown in Fig. 4, we observe that the convergence rate of the proposed PCFADFE is better than PIPEADFE1 and PIPEADFE2. That is, the convergence performance is improved by the introduced Post-cursor processing filter. The

Authorized licensed use limited to: National Taiwan University. Downloaded on March 2, 2009 at 02:33 from IEEE Xplore. Restrictions apply.

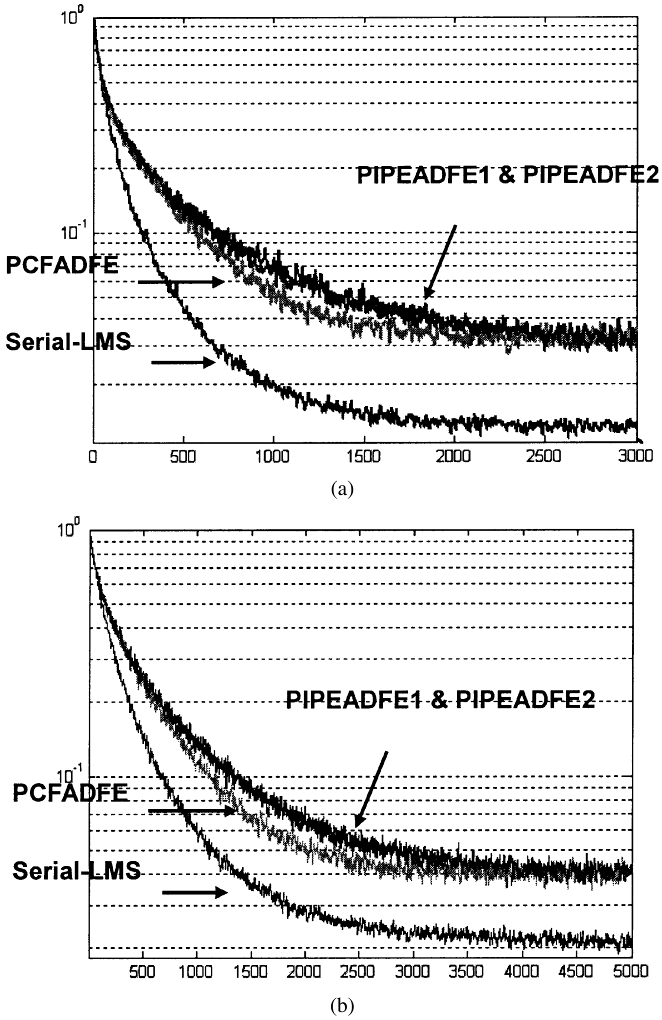

Fig. 4. Learning curve of PIPEADFE1, PIPEADFE2, and PCFADFE for (a) Channel I and (b) Channel II.

effect of PCF can be explained as follows. It is well known that the convergence rate of the conventional LMS-based ADFE depends on the step size and the channel spectral characteristics, which are related to the eigenvalue (λ_n) of the received signal autocorrelation matrix [10]. If the channel amplitude and phase distortions are small, the eigenvalue ratio $(\mathrm{Max}(\lambda_n)/\mathrm{Min}(\lambda_n))$ is close to unity and, the ADFE converges to its optimal tap coefficients relatively fast. On the other hand, if the channel exhibits poor spectral characteristics, such as relatively large attenuation in part of its spectrum, the eigenvalue ratio $\mathrm{Max}(\lambda_n)/\mathrm{Min}(\lambda_n) \gg 1$. Then, the convergence rate of LMS-based DFE will be slow. On the other hand, by using the PCF, the decisions or the training sequences can be applied to the updating mechanism, and the eigenvalue spread of input signal should be reduced. Thus, the convergence rate of PCFADFE can be faster than the PIPEADFE1. Moreover, the PCFADFE performs better than PIPEADFE2 since it provides more degrees of freedom in updating the PCF than the PP unit of PIPEADFE2.

TABLE II
HARDWARE COMPLEXITY OF PIPEADFE1, PIPEADFE2, AND PCFADFE

	D_l	1	N
PIPEADFE1	Mult. in FFF and PCF	$2N_f$	$2N_f$
	Mult. in FBF	$2N_b$	$2N_b$
	Total Adder	$2N_f+2N_b$	$2N_f+2N_b$
PIPEADFE2	Mult. in FFF and PP	$2N_f+1$	$2N_f+N$
	Mult. in FBF	$2N_b$	$2N_b$
	Total Adder	$2N_f+2N_b+1$	$2N_f+2N_b+N$
PCFADFE	Mult. in FFF and PCF	$2N_f+2$	$2N_f+2N$
	Mult. in FBF	$2N_b$	$2N_b$
	Total Adder	$2N_f+2N_b+2$	$2N_f+2N_b+2N$

B. Comparison of Hardware Complexity

According to our simulation PIPEADFE1, PIPEADFE2 and PCFADFE will converge to the same SNR at the same N_f, N_b, D_1 and step size μ. The D_1 versus hardware complexity is shown in Table II. It can be seen that the hardware complexity of PIPEADFE2 and PCFADFE are close. Nevertheless, the convergent rate of the proposed PCFADFE is improved than PIPEADFE1 and PIPEADFE2.

V. Conclusions

In this paper, a new pipelined ADFE using the PCF technique is presented. Compared with the algorithm in [6], we show the convergence rate of the proposed algorithm can be improved, while the hardware overhead is close to the VLSI architecture in [6]. We demonstrated the effectiveness of the new design methodology by simulations. It provides an alternative approach for the design of high-seed pipelining ADFE with arbitrary speedup factor.

References

[1] J. G. Proakis, *Digital Communications*, 4th ed. New York: McGraw-Hill, 2001, pp. 677–678.

[2] E. A. Lee and D. G. Messerschmitt, *Digital Communication*, 2nd ed. Norwell, MA: Kluwer, 1994, pp. 518–548.

[3] M. A. Soderstrand, A. E. de la Serna, and H. H. Loomis Jr, "New approach to clustered look-ahead pipelined IIR digital filters," *IEEE Trans. Circuit Syst. II*, vol. 42, pp. 269–274, Apr. 1995.

[4] K. K. Parhi, *VLSI Digital Signal Processing System*. New York: Wiley, 1999.

[5] ——, "Pipelining in algorithm with quantizer loops," *IEEE Trans. Circ. Syst.*, vol. 38, pp. 745–754, July 1991.

[6] N. R. Shanbhag and K. K. Parhi, "Pipelined adaptive DFE architectures using relaxed look-ahead," *IEEE Trans. Signal Processing*, vol. 43, pp. 1368–1385, June 1995.

[7] M. Schobinger *et al.*, "CMOS digital adaptive decision feedback equalizter chip for multilevel QAM digital radio modems," in *Proc. IEEE Int. Symp. Cir. Syst.*, vol. 28, Mar. 1993, pp. 330–338.

[8] G. Long, F. Ling, and J. G. Proakis, "The LMS algorithm with delayed coefficient adaptation," *IEEE Trans. Acoust. Speech Signal Processing*, vol. 37, pp. 1397–1405, Sept. 1989.

[9] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

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

Authorized licensed use limited to: National Taiwan University. Downloaded on March 2, 2009 at 02:33 from IEEE Xplore. Restrictions apply.