A Systolic Architecture for LMS Adaptive Filtering with Minimal Adaptation Delay

S. Ramanathan
raman@cadl.iisc.ernet.in

V. Visvanathan
vish@cadl.iisc.ernet.in

Supercomputer Education and Research Centre
Indian Institute of Science
Bangalore - 560 012, India

Abstract

Existing systolic architectures for the LMS algorithm with delayed coefficient adaptation have large adaptation delay and hence degraded convergence behaviour. This paper presents a systolic architecture with minimal adaptation delay and input/output latency, thereby improving the convergence behaviour to near that of the original LMS algorithm. The architecture is synthesized by using a number of function preserving transformations on the signal flow graph representation of the delayed LMS algorithm. With the use of carry-save arithmetic, the systolic folded pipelined architecture can support very high sampling rates, limited only by the delay of a full adder.

1 Introduction

The use of delayed coefficient adaptation in the LMS algorithm has enabled the design of modular systolic architectures for real-time transversal adaptive filtering [1]-[3]. However, the convergence behaviour of this delayed least mean squares (DLMS) algorithm, when compared with that of the standard LMS algorithm, is degraded and worsens with the increase in the adaptation delay [4]. Large number of adaptation delays have been used in previous systolic architectures, since they have been necessary for systolizing the system to support high sampling rates in a real-time environment. Hence, the design of a modular systolic architecture for transversal adaptive filtering, that maintains the convergence behaviour of the LMS algorithm by minimizing the adaptation delay, and also supports high input sampling rates with minimal input/output latency, is an important problem [5].

The area-efficient modular systolic architecture derived in this paper, uses a number of function preserving transformations to modify the standard signal flow graph (SFG) representation of the DLMS algorithm into a systolic architecture with minimal adaptation delay (D_A). The key transformations used are slowdown and folding [6], which were also used in [3] to reduce D_A to half that of [1] and [2]. Here, we generalize this idea to develop a systolic architecture, where slowdown by an arbitrary factor P, results in a proportionate reduction in D_A. A further reduction in D_A is achieved by use of the associativity of addition. This idea is adapted from [4], where a semi-systolic architecture is presented wherein the error is broadcast to all the weight update units. Here, we ensure a scalable systolic architecture by limiting the number of fan-out lines (denoted by M) in each systolic processor, so that the broadcast delay is less than the system clock period (T_c). This further reduces D_A by a factor of approximately M. Finally, as in [3], the use of associativity of addition reduces the input/output latency to a small constant, which is independent of the filter order. With the use of carry-save arithmetic, the systolic folded pipelined architecture can support very high sampling rates, limited only by the delay of a full adder.

The organization of this paper is as follows. In Section 2 we derive the systolic architecture. In Section 3 we analyse and evaluate the architecture and present simulation results of an adaptive line enhancer that shows the improved convergence behaviour due to reduced D_A. We summarize the work in Section 4.

2 Deriving the Systolic Architecture

The architecture shown in Figure 1 is derived from the standard SFG representation of the DLMS algorithm [1] by applying the holdup, associativity, retiming, and slowdown transformations [6], in that order. In Figure 1, D_H is the holdup delay, P is the slowdown factor, N is the filter order, μ is the step-size, and, X, Y and R are respectively, the input, the output and the desired response. There are MP fan-out lines from a register in the error broadcast path. For ease of exposition, we assume N to be a multiple of MP. Note that, in the retiming step, $\frac{N}{MP}$ registers

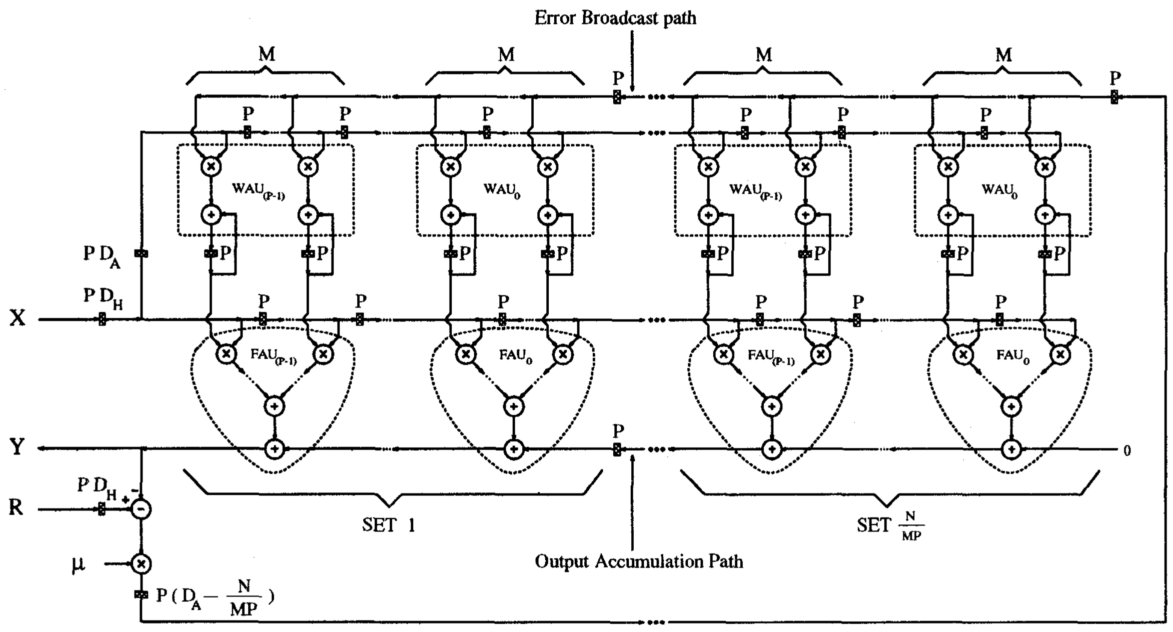

Figure 1: Systolic slowed-down DLMS adaptive filter architecture.

have been moved to break the error broadcast path and the output accumulation path at regular intervals. This retiming is possible due to the use of the adaptation delays. The system inputs are assumed to be P-slow [6], and hence the hardware utilization efficiency of this architecture is $\frac{100}{P}\%$.

Note that, in Figure 1 $\frac{N}{MP}$ sets of arithmetic units have been identified. In each set, the P arithmetic units in the filtering and weight-update portions are denoted as filtering arithmetic units (FAUs) and weight-update arithmetic units (WAUs) respectively. In order to regain 100% HUE, we use the folding transformation [6]. Using locally-sequential-globally-parallel mapping, the P FAUs and WAUs of a set are mapped onto one physical FAU and WAU respectively. Further, the FAUs and the WAUs are scheduled in the order of their index j, $j = 0, \ldots, (P-1)$. The control circuitry of the resulting folded architecture is derived using the approach reported in [6]. Figure 2 shows the systolic array, while the details of the boundary processor module (BPM) and a folded processor module (FPM) are shown in Figure 3.

The complex control circuitry present at the input to the FAU and WAU of an FPM (refer Figure 3(b)), consisting of a delay-line with $(MP-1)P$ registers and P-to-1 multiplexers Mux i, i=1, ..., M, is replaced by a simple structure (refer Figure 4(b)) consisting of a 2-to-1 multiplexer in a loop along with $(MP-1)$ registers. The correctness of this structural optimization is established in [7]. Further, by moving appropriate number of registers from the BPM into the FPMs, the processors are pipelined as shown in Figure 4. Using the standard notation [6], the pipeline registers are shown at the output of the arithmetic units. The number of pipeline registers p_1 and p_2 are given by: $p_1 = \left\lceil \frac{T_m}{T_c} \right\rceil$ and $p_2 = \left\lceil \frac{T_m + \lceil \log_2 M \rceil T_a}{T_c} \right\rceil$, where, T_m and T_a denote the delay of a multiplier and an adder respectively. For ease of exposition, we have neglected the delay associated with the pipeline registers. Since, pipelining changes the multiplexer definitions, an extended retiming theorem presented in [7] is used to redefine them. The precise multiplexer definitions are available in [7]. Note that, in an FPM, the error signal is broadcast from a synchronization register to M

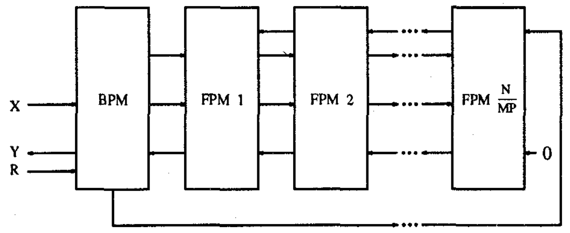

Figure 2: Folded systolic array for DLMS algorithm.

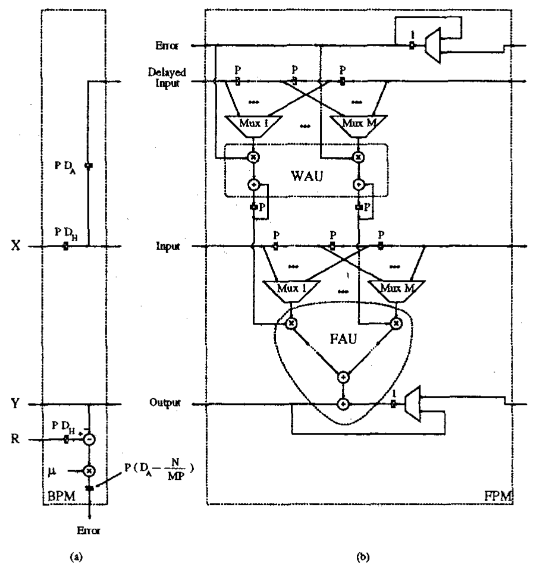

Figure 3: (a) Boundary Processor Module. (b) Folded Processor Module.

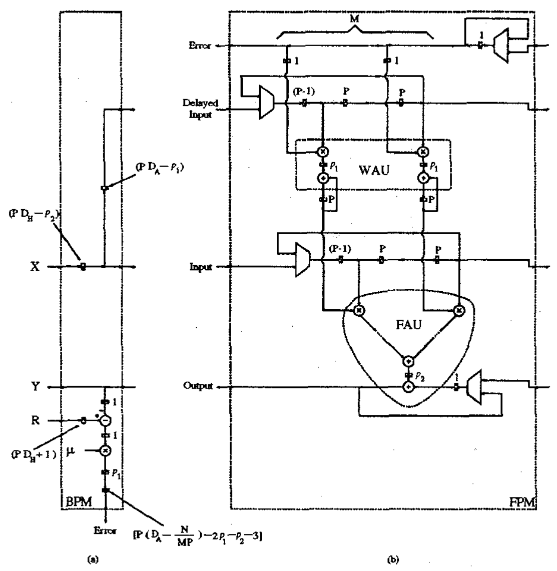

Figure 4: (a) Final Boundary Processor Module. (b) Final Folded Processor Module.

holding registers at the inputs to the multipliers in the WAU. This is done in order to minimize the delay associated with the broadcast and thereby maximize the value of M that can be used with any given clock period. The final BPM and FPM are shown in Figure 4, while the complete systolic array is as shown in Figure 2. Note from Figure 4(a), the minimum D_A and D_H are given by:

$$D_A = \frac{N}{MP} + \left\lceil \frac{2p_1 + p_2 + 3}{P} \right\rceil \quad (1)$$

$$D_H = \left\lceil \frac{p_2}{P} \right\rceil \quad (2)$$

The performance of the architecture derived above is analyzed and evaluated in the following section.

3 Analysis and Evaluation

In Table 1, we compare the new architecture with the systolic architecture reported in [2]. It should be noted that the performance metrics of [1] are essentially the same as that of [2]. From the table, it is clear that the new architecture gains over [2] by a factor MP in the adaptation delay and by a factor P in hardware requirements, with minimal control overheads. Also, the input/output latency is independent of N. Further, the fastest sampling rate that the new architecture can support is $\frac{1}{T_a}$.

The degrading convergence behaviour of the DLMS algorithm with increase in D_A is verified through the

Comparison Metrics	[2]	New Architecture
Min. T_s	$2(T_m + T_a)$	T_a
D_A	N	$\frac{N}{MP} + \left\lceil \frac{2p_1+p_2+3}{P} \right\rceil$
D_H	N	$\left\lceil \frac{p_2}{P} \right\rceil$
# of Multipliers	$3N$	$\frac{2N}{P} + 1$
# of Adders	$(2N + 1)$	$\frac{2N}{P} + 1$
# of Multiplexers	-	$\frac{4N}{MP}$

Table 1: Comparison of Systolic DLMS Architectures.

adaptive line enhancer (ALE) example. The input to the 64-tap ALE consists of two sine waves of power and frequency $\sigma_1^2 = 0.5$, $f_1 = 1$MHz, and $\sigma_2^2 = 0.5$, $f_2 = 2.5$MHz, corrupted by an independent broadband additive gaussian noise of power $\sigma_n^2 = 0.1$. The sampling frequency is 10MHz, the decorrelation delay Δ is 1, and the Weiner error for this setting is 0.1065. Figure 5 shows the simulation results obtained by averaging over 1000 runs with $D_A = 0, 8, 16, 32$, and 64. In each case the step-size was chosen to get fast critically damped convergence.

It is clear from Figure 5, that a smaller D_A results in convergence speed that is closer to that of the LMS algorithm. From eqn. (1), for a given sampling period (T_s), the use of a large P (where, $P = \lceil \frac{T_s}{T_c} \rceil$) results in a small D_A. This then translates to the use of a very fast system clock. The critical paths restricting the clock speed are the weight update loop and the output accumulation loop in the FPMs (refer Fig-

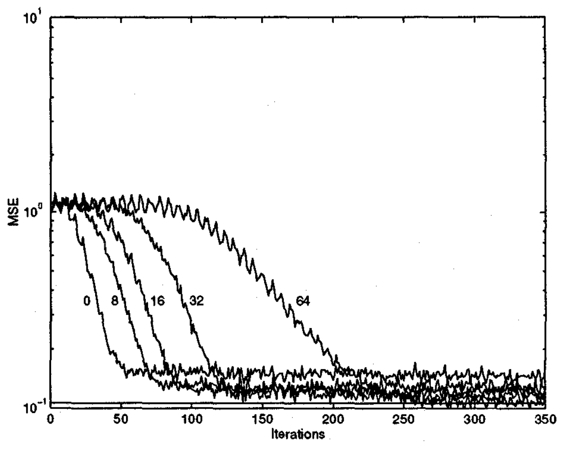

Figure 5: Convergence plots of ALE example for various values of adaptation delay.

ure 4(b)). These updates and accumulations can be done in carry-save form. In the case of the weight update loop, the vector merging can be done immediately outside the loop, while in the case of the output accumulation loop, the entire accumulation can be kept in carry-save form with vector merging done in the BPM. Further, if these vector merge adders are suitably pipelined, then the critical path in the system consists of a full adder and a 2-to-1 multiplexer. In present day technology, this delay is of the order of a nanosecond, and hence the clock speed is in practice limited only by the ability to generate and distribute a very fast clock. Thus, reasonable values of P and hence a significant reduction in D_A can be achieved even for very high input sampling rates, say of the order of a hundred mega samples per second. Note that, use of a fast clock results in a small M.

Even if other system design considerations preclude the use of a very fast clock for a given T_s, the required D_A is only marginally more than that obtainable with a very fast clock. To get a better insight into this statement, assume a linear relationship between the error broadcast delay (T_b) in an FPM and the total number of fan-out lines (M), ie., $T_b = T_f M$, where, T_f is the delay per fan-out line. For the correct operation of the architecture, we require $T_b \leq T_c$. Therefore, the largest possible value of M is $\frac{T_c}{T_f}$ and hence $MP = \frac{T_s}{T_f}$. Hence, the first term in the expression for D_A (eqn. (1)) is proportional to the computational throughput of the system and is independent of T_c, while, the second term has only a logarithmic dependence on M. Note however, due to a slow clock and therefore a small P, the savings in area degrades proportionately with P.

It is clear from the above discussion that reasonably high values of MP can be obtained for many applications. This, as is substantiated by the ALE example, results in a convergence speed that is close to that of the LMS algorithm and is significantly superior to that of [1] and [2].

4 Summary

We have presented an area-efficient modular systolic architecture for DLMS adaptive filtering with minimal adaptation delay and input/output latency. Due to the significant reduction in the adaptation delay, the convergence behaviour of the proposed architecture is considerably closer to that of the LMS algorithm than the architectures of [1] - [3]. The architecture was synthesized by the use of a proper sequence of function preserving transformations, namely associativity, retiming, slowdown, and folding. With the use of carry-save arithmetic, the systolic folded pipelined architecture can support very high sampling rates, limited only by the delay of a full adder.

References

[1] H. Herzberg, R. Haimi-Cohen, and Y. Be'ery "A systolic array realization of an LMS adaptive filter and effects of delayed adaptation," *IEEE Trans. Signal Processing*, vol. 40, no. 11, pp. 2799-2802, Nov. 1992.

[2] M. D. Meyer and D. P. Agrawal, "A high sampling rate delayed LMS filter architecture," *IEEE Trans. Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 40, no. 11, pp. 727-729, Nov. 1993.

[3] V. Visvanathan and S. Ramanathan, "A modular systolic architecture for delayed least mean squares adaptive filtering," *Proc. Intl. Conf. VLSI Design*, New Delhi, pp. 332-337, Jan. 1995.

[4] G. Long, F. Ling, and J. G. Proakis, "The LMS algorithm with delayed coefficient adaptation," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 37, no. 9, pp. 1397-1405, Sept. 1989, and corrections, vol. 40, no. 1, pp. 230-232, Jan. 1992.

[5] 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-II: Analog and Digital Signal Processing*, vol. 40, no. 12, pp. 753-766, Dec. 1993.

[6] K. K. Parhi, C-Y. Wang, and A. P. Brown, "Synthesis of control circuits in folded pipelined DSP architectures," *IEEE J. Solid State Circuits*, vol. 27, no. 1, pp. 29-43, Jan. 1992.

[7] S. Ramanathan and V. Visvanathan, "A systolic architecture for DLMS adaptive filtering," *Technical Report*, Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore, 1995.