Available online at www.sciencedirect.com
ScienceDirect
INTEGRATION, the VLSI journal 40 (2007) 209–225
INTEGRATION the VLSI journal
www.elsevier.com/locate/vlsi

Design of efficient architectures for 1-D and 2-D DLMS adaptive filters

K.R. Santha[a,*], V. Vaidehi[b]

[a]Department of Electrical and Electronics Engineering, Sri Venkateswara College of Engineering, Chennai
[b]Department of Electronics, Madras Institute of Technology, Chennai

Received 3 September 2004; received in revised form 18 September 2005; accepted 24 October 2005

Abstract

This paper presents a design of systolic array architectures for 1-D and 2-D finite impulse response adaptive filters. The design is based on the delayed least mean squares algorithm (DLMS). Two designs have been proposed and their performance is analyzed in terms of speed up, adaptation delay and throughput. The second realization results in a lowest critical period equal to one multiply operation time and a higher throughput. Unlike the existing architectures the adaptation delay of this design is independent of the filter length and has a higher speedup. It is shown that the convergence performance of this design is in par with the conventional LMS algorithm.

Keywords: Adaptive filters; 2-D DLMS algorithm; Systolic architectures; Equalization

1. Introduction

Adaptive filters have a wide range of applications in wireless communications and image processing. Hence the need of high-speed architectures for such systems for use in high data-rate applications is essential. This has been the topic of research over a long period. Pipelining along with parallel processing are the two design technologies for such high-speed applications. Systolic array designs, in particular, are receiving much attention for dedicated VLSI processors. Systolic arrays are a network of several interconnected identical cells which satisfy the constraints of locality, modularity and pipelinability. The attractive feature of this array is that data can be pumped regularly through the system, thus providing a uniform and fast throughput rate. In this paper the research is focused on reducing the critical period thereby increasing the clock-rate and throughput. The most widely used adaptation algorithm is the least mean squares (LMS) algorithm [1] because of its simple algorithmic complexity and robust performance. The pipelined and/or parallel implementations of the LMS adaptive filter introduce delays in adaptation. Latches have to be introduced to implement the adaptation process and this leads to the delayed least mean squares (DLMS) algorithm. The delayed adaptation degrades the performance of the system when compared to the systems employing the LMS algorithm. This delay also increases in proportion to the order of the filter. Hence the hardware implementation of the LMS algorithm that minimizes the amount of adaptation delay and makes it independent of the order of the filter is of great interest. Pipelined implementations have been reported in [2] and [3] which exhibit no adaptation delay. However these architectures lead to complex circuits and the hardware overhead is large. The binary tree structure proposed in [4,5] achieves a delay of $\log_2 N + 1$ cycles where N is the order of the filter. This is less than N but the structure is not purely systolic. Pure systolic designs have been presented in [6] and [7] where the delays are $N-1$ and N, respectively. But the critical periods are very high which make the systems slow. Latest research is reported in [8] which is more or less the same as that of [4] for $N = 2^x$ where x is an integer. Thus Van et al. have proposed a systolic architecture that has the lowest critical period which is one multiply–add operation time and a minimum delay. However this work proposes a tree-systolic processing element. In all these schemes the delays depend on N. As a result, in this paper an attempt is made

*Corresponding author. Tel.: +91 044 24514649; fax: +91 041 11262462.
E-mail addresses: santha@svce.ac.in (K.R. Santha), vaidehi@mitindia.edu (V. Vaidehi).

0167-9260/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.vlsi.2005.10.003

to reduce the critical period with an optimum delay and make it independent of the filter order.

The works in [2,8] have been utilized to design 2-D DLMS filters and reported in [9,10] respectively. In [9] a method is proposed to reduce the delay by reversing the direction of output. However this uses the look-ahead technique increasing the hardware complexity. Taking into account of all these reasons two methods have been proposed for 1-D and 2-D DLMS FIR filters. The conventional 1-D LMS algorithm is chosen and the systolic array is developed using the 'Systolic Array Design Algebra' proposed by Kung [11]. This is further extended to develop systolic architectures for 2-D DLMS FIR filters [12–14]. The second method achieves a lowest critical period of one multiply operation time and the adaptation delay remains to be the lowest of all the existing schemes. It is a constant independent of the order of the filter. With the trend towards increasingly high transmission rates, there is a tremendous need for high-speed equalizer architectures. The parallel-pipelined 2-D architectures can perform faster when compared to 1-D architectures with the same number of multipliers and adders. The 2-D architecture is particularly useful in image processing applications. Here the convergence results of the 1-D filter are analyzed assuming an adaptive equalization set up. The performance of the 2-D filter is also validated considering the noise removal in images. The functionality and timing has been verified by simulation via ACTEL'S VERIBEST VHDL simulator. The synthesis results are also presented. The rest of the paper is organized as follows. Section 2 explains the design of the array architectures. Section 3 presents the comparison and implementation results. The applications are discussed in Sections 4 and 5 presents conclusions.

2. Design of the architecture

The conventional 1-D LMS algorithm is given as

$$y(n) = \sum_{k=0}^{N-1} w_K(n)x(n-k), \tag{1}$$

$$e(n) = d(n) - y(n), \tag{2}$$

$$w_K(n+1) = w_K(n) + \mu e(n)x(n-k), \tag{3}$$

where $k = 0, 1, 2, \ldots, N-1$; $x(n)$ and $w_K(n)$ represent the nth input sample and the kth tap weight for an N-tap adaptive filter, respectively. $y(n)$, $d(n)$, $e(n)$ and μ denote the filter output, the desired input, the error and the adaptation step size, respectively. The weight update equation using the delayed LMS algorithm with D as the amount of adaptation delay is given as

$$w_K(n+1) = w_K(n) + \mu e(n-D)x(n-k-D). \tag{4}$$

2.1. The proposed architecture I

A systolic architecture for the LMS algorithm is derived by utilizing the algebra of [11]. Fig. 1 shows the z-graph representation of the algorithm given by Equations (1)–(4) for an N-tap filter. Each node a_i and b_i represent a multiply-add operation. To enable high speed operation each of the multiply, add operations can be parallelized. By doing so during each clock, four operations occur in parallel in each node. This is achieved by introducing a delay of one unit between the multiplier and adder in the node or processor of Fig. 1. The z-graph representation for this operation is given in Fig. 2. Here m_i and n_i represent the multiplication nodes, a_i and b_i represent the addition nodes. Combining these nodes the short form I of the z-graph is obtained as in Fig. 3. The symbolic state-space model of the short form I for the modified z-graph is obtained by representing the filter and update operations as a single node variable V_i as in Fig. 4. The input–output relation of each node is the state model given by Eqs. (5)–(7).

$$V(n) \leftarrow AV(n)' + 'BI(n), \tag{5}$$

$$y(n-1) = C^{\mathrm{T}}V(n), \tag{6}$$

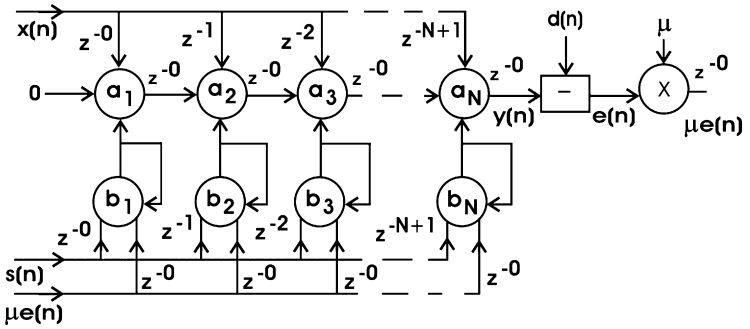

Fig. 1. Z-graph representation of the conventional LMS algorithm.