International Journal of Computer and Communication Technology

Volume 3 | Issue 4

Article 15

October 2012

VLSI Design and Implementation for Adaptive Filter using LMS Algorithm

Asit Kumar Subudhi Asst. Prof., DRIEMS, Cuttack, asit82.subudhi@gmail.com

Biswajit Mishra ITER, Siksha O' Anusandhan University, Bhubaneswar, biswajitmishra@gmail.com

Mihir N. Mohanty Asst.Prof. H.O.D,ITER,SOAU, Bhubaneswar, mihir.n.mohanty@gmail.com

Follow this and additional works at: https://www.interscience.in/ijcct

Recommended Citation

Subudhi, Asit Kumar; Mishra, Biswajit; and Mohanty, Mihir N. (2012) "VLSI Design and Implementation for Adaptive Filter using LMS Algorithm," *International Journal of Computer and Communication Technology*: Vol. 3 : Iss. 4 , Article 15. DOI: 10.47893/IJCCT.2012.1160 Available at: https://www.interscience.in/ijcct/vol3/iss4/15

This Article is brought to you for free and open access by the Interscience Journals at Interscience Research Network. It has been accepted for inclusion in International Journal of Computer and Communication Technology by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.

VLSI Design and Implementation for Adaptive Filter using LMS Algorithm

Asit Kumar Subudhi¹, Biswajit Mishra², Mihir Narayan Mohanty^{3*} ¹Asst. Prof., DRIEMS, Cuttack^{2,3}ITER, Siksha O' Anusandhan University, Bhubaneswar ^{1*}asit82.subudhi@gmail.com, ^{3*}mihir.n.mohanty@gmail.com

Abstract Adaptive filters, as part of digital signal systems, have been widely used, as well as in applications such as adaptive noise cancellation, adaptive beam forming, channel equalization, and system identification. However, its implementation takes a great deal and becomes a very important field in digital system world. When FPGA (Field Programmable Logic Array) grows in area and provides a lot of facilities to the designers, it becomes an important competitor in the signal processing market. In general FIR structure has been used more successfully than IIR structure in adaptive filters. However, when the adaptive FIR filter was made this required appropriate algorithm to update the filter's coefficients. The algorithm used to update the filter coefficient is the Least Mean Square (LMS) algorithm which is known for its simplification, low computational complexity, and better performance in different running environments. When compared to other algorithms used for implementing adaptive filters the LMS algorithm is seen to perform very well in terms of the number of iterations required for convergence. This phenomenon can be achieved by a sufficient choice of bit length to represent the filter's coefficients. This paper presents a lowcost and high performance programmable digital finite impulse response (FIR) filter. It follows the adaptive algorithm used for the development of the system. The architecture employs the computation sharing algorithm to reduce the computation complexity.

Introduction

The FIR filter computations play an important role in the DSP systems as the adaptive type. The FIR filtering is a convolution operation which can be viewed as the weighted summation of the input sequences. The large amount of multiply-and accumulate operations attracts many researchers to study this issue. The FIR Filter consists of shifters, adders, and multipliers. These constitutes can be chosen on the demand of the designer. One of the most popular adaptive algorithms available in the literature is the stochastic gradient algorithm also called least-mean-square (LMS) [1], [2]. Its popularity comes from the fact that it is very simple to be implemented. As a consequence, the LMS algorithm is widely used in many applications. Least - Mean - Square (LMS) adaptive filter is

the main component of many communication systems; traditionally, such adaptive filters are implemented in Digital Signal Processors (DSPs). Sometimes, they are implemented in ASICs, where performance is the key requirement. However, many high-performance DSP systems, including

LMS adaptive filters, may be implemented using Field Programmable Gate Arrays (FPGAs) due to some of their attractive advantages. Such advantages include flexibility and programmability, but most of all, availability of tens to hundreds of hardware multipliers available on a chip [3]. The LMS adaptive filter enjoys a number of advantages over other adaptive algorithms, such as robust behavior when implemented in finite-precision hardware, well understood convergence behavior and computational simplicity for most situations as compared to least square methods [4].

Identifying an unknown system has been a central issue in various application areas such as control, channel equalization, echo cancellation in communication networks and teleconferencing etc. Identification is the procedure of specifying the unknown model in terms of the available experimental evidence, that is, a set of measurements of the input output desired response signals and an appropriately error that is optimized with respect to unknown model parameters. Adaptive identification refers to a particular procedure where we learn more about the model as each new pair of measurements is received and we update the knowledge to incorporate the newly received information. The Least Mean Square (LMS) adaptive filter is a simple well behaved algorithm which is commonly used in applications where a system has to adapt to its environment. Architectures are examined in terms of the following criteria: speed, power consumption and FPGA resource usage [6]. Modern FPGAs contain

International Journal of Computer and Communication Technology (IJCCT), ISSN: 2231-0371, Vol-3, Iss-4

many resources that support DSP applications .These resources are implemented in the FPGA fabric and optimized for high performance and low power consumption [7].

This paper proposes a VHDL implementation of a Least Mean Square (LMS) adaptive algorithm. The good convergence of LMS algorithm has made us to choose it. It also has good stability. Adaptive filtering constitutes one of the core technologies in digital signal processing and finds numerous application areas in science as well as in industry. In this paper LMS algorithm is used to reduce the error at the output of the system. A VHDL implementation is developed for a LMS adaptive filter. It has been proven that LMS Algorithm has good behavior. **Canonical Sign Digit Algorithm (CSD)**

The CSD representation is a radix-2 signed digit system with digit set {-, 1, 0}, where '-' represent '-1'. For a given constant, the corresponding CSD representation is unique and has two properties; the first is that the number of non-zero digits is minimum and the second is that the product of adjacent two digits is zero, that is, two non-zero digits are not adjacent[2].

In CSD, an integer X can be represented by,

$$X = \sum_{i=0}^{n} C_i 2^i$$

where, $C_i \in \{-, 1, 0\}$. For example, 255 is 1111111 in binary representation and 10000001 in CSD.

It is signed digit number system that minimizes the number of non-zero digits .It can reduce the number of partial product additions in a hardware multiplier. They are successful in implementing multipliers with less complexity. Since the complexity of the multipliers is typically estimated through the number of non-zero elements, which can be reduced by using signed digit numbers. Adjacent CSD digits are never both non-zero. This property implies that for an n-bit number, there are at most [n/2] non-zero digits.

For a 2's complement number there can be n non-zero digits for n-bit number. The probability of a digit being zero is roughly 2/3 for CSD and exactly 1/2 for 2's complement. For negative numbers, the numbers of non-zero digits is less for the CSD Representation than the 2's Complement representation. The CSD numbers has the minimum number of non-zero digits and no consecutive nonzero digits[6] .The CSD Representation have fewer nonzero digits than the normal binary expression. Now the multipliers in the digital filters are realized with shifters, adders and subtractors. The use of CSD expression can reduce the number of adders and subtractors For example, the normal binary representation would need 3 adders, as 15 is represented as 11112. The number of adders and subtractors is less than the number of non-zero digits by 1.The CSD Multiplier is based on shifts and adds (or subtracts) instead of conventional multipliers. This results in the area reduction of multiplier of the digital filters. The Complexity of a digital filter design is a function of the number of non-zero design in the filter Coefficients. Encoding the filter Coefficient using the CSD representation reduces the number of partial products as well as the area and the power consumption. Hence, it is useful technique for implementing of FIR Filters with fixed coefficients. Digital FIR filters can take advantage of the CSD representation.

The Coefficient of Low pass FIR Filter is represented in signed and unsigned numbers. Two's complement arithmetic efficiently handles the addition and multiplications of signed numbers. In DSP filtering applications, coefficients are made up of both positive and negative numbers. Depending on the applications data is either positive or negative. Two's complement arithmetic efficiently handles the addition and multiplications of signed numbers. The advantages of two's complement are that we can use the same hardware to add negative numbers and positive numbers and the carry out is discarded. So the Coefficient firstly convert into two's Complement then CSD Algorithm is applied which convert the representation into maximum number of non-zero term. There is a flowchart which will convert the two's Complement number into CSD representation shown below[1].

A filter represented by a CSD code is called CSD filter. The multiplication can be easily implemented by using CSD code coefficients. The number of adders/subtractors required to realize a CSD multiplier is one less than the number of nonzero digits in the CSD code.

CSD Multiplier

The multiplier may contain set of coefficients. The Coefficients are multiplied with operands that are supplied to multiplier. Each operand may be multiplied by a coefficient. Conventional multipliers are generally implemented with combinations of shift and add operations. Some of such implementations result in inefficient use of area. The simulation result is shown in result section figure 1.

LMS Algorithm

LMS is the most widely used algorithm. The key feature of the LMS algorithm is its simplicity. It requires neither measurement of the correlation function, nor matrix inversion [4] .It uses Mean Square Error (MSE) as a criterion. LMS uses a stepsize parameter, input signal and the difference of desired signal and filter output signal to frequently calculate the update of the filter coefficients set [3].

1) LMS Equation: The simplest estimation may use only the current available taps and the current desired response to estimate the autocorrelation matrix and the cross-correlation vector. The equation to adapt tap weights w(n) using the instantaneous taps x(n) and desired response d(n) is [1]:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mathbf{x}(n) [\mathbf{d}(n) - \mathbf{x}(n)\mathbf{w}(n)]$$
(1)

where is the step size, since the filter output is the convolution sum of the taps and tap weights

$$\mathbf{y}(\mathbf{n}) = \mathbf{x}(\mathbf{n})\mathbf{w}(\mathbf{n})$$
(2)

and the estimated error signal e (n) is defined as the difference between the desired response and the filter response,

or,
$$e(n) = d(n) - y(n)$$
 (3)

So, Eq. (1) can be rewritten in terms of the error signal and the taps:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mathbf{x}(n) \mathbf{e}(n)$$
(4)

Eq. (4) is the formula for the LMS algorithm [5]. As illustrated in the equation, each tap weight adaptation at each time interval requires merely the knowledge of the current taps and the current error signal, which is produced with the knowledge of the desired response. The algorithm does not require any prior knowledge of the entire autocorrelation matrix or the cross-correlation vector, nor does it require matrix computations.

2) The Convergence boundary: The convergence time of the LMS algorithm depends on the step size

. If is small, then it may take a long convergence time and this may defeat the purpose of using an LMS filter. However if is too large, the algorithm may never converge. LMS algorithm can be shown to converge for values of less than the reciprocal of the largest eigenvalue of the autocorrelation matrix of x(n), but it may be time-varying.

3) Selection of Adaptive Parameters: The choice of the step-size parameter and the order of the filter effectively determines the performance of LMS [3]. When the filter taps are increased, this improves the convergent performance of LMS algorithm, but every tap (in structure of LMS adaptive filter) costs two more multipliers and two more adders as seen in Figure 3. However, this will increase the area needed and decrease the maximum frequency of the design. So, balance is required between the convergent performance and the amount of hardware used effectively. Unfortunately, there is no clear mathematical analysis to derive the exact quantities. Only through experiments may a reasonable solution be obtained [3]. In order to select appropriate step size and filter order, MATLAB simulation of LMS algorithm is carried out. Based on the simulation results (which will be discussed later), the adaptive parameters obtained will be applied to the hardware implementation process of LMS algorithm.

VHDL Implementation of LMS Algorithm

1) Processing of positive and negative number: In FPGA, the data, such as the input signals, the coefficients of filter, and the desired signal, may be positive or negative. So it is necessary to use signed numbers for expressing all the data inside FPGA.

International Journal of Computer and Communication Technology (IJCCT), ISSN: 2231-0371, Vol-3, Iss-4

Based on the expression method of the signed number, the MSB bit is used as a sign bit. For the MSB bit, binary digit '0' denotes a positive number, binary digit '1' expresses a negative number, and all the data are denoted by the 2's complement.

2) Fixed-point representation of Data: Fixed point is a step between integer mathematics and floatingpoint. This has the advantage of being almost as fast as integer arithmetic, and able to represent numbers with fraction. It uses a smaller area in FPGA than floating-point to process the arithmetic operations. A fixed-point number has an assigned width and an assigned location for the decimal point. As long as the number is big enough to provide enough precision, fixed point is fine for most DSP applications. Because it is based on integer math, it is extremely efficient as long as the data does not vary too much in magnitude.

The most essential task is the right selection of word lengths for the various variables in the system. Short word lengths may result in extra roundoff errors, which can cause instability or poor performance. On the other hand, the use of excessively long word lengths increases system complexity which in turn reduces its maximum speed and increases the used area of the FPGA. So balance should be achieved between the system round-off errors and the maximum speed of operation together with the used area of the FPGA [8].

In order to select a sufficient word length and a range for the various variables in the system, we should design the proposed adaptive FIR filter according to the application (adaptive noise canceller) . So, a MATLAB simulation model of fixed-point adaptive noise canceller is used in order to get the variables' representation with minimum possible word length and sufficient partitioning between integer part and decimal fraction, which results from model tests with varying word lengths and decimal fractions [6]. A bad choice of a decimal fraction produces more quantization noise. Table 1 shows the numerical ranges of the input signals, the output signal, and the coefficients obtained by simulation. Based on the ranges obtained, the locations of the bits for the integer parts and those for the decimal fractions of each variable are obtained.

Result and Conclusion

A review of adaptive filters shows that the LMS algorithm is still a popular choice for its stable performance and high speed capability. The other advantage of the LMS over other adaptive algorithm is its high convergence rate. The high-speed capability and register rich architecture of the FPGA is ideal for implementing LMS. A hybrid adaptive filter is designed with a direct-form FIR filter coded in VHDL and with the LMS algorithm written in VHDL code executing on the Xilinx output is simulated on MATLAB.

A conclusion of the performance of the LMS adaptive filtering algorithm is expressed by its simplicity to implement and its stability when the step size parameter is selected appropriately. This made the LMS algorithm the acceptable choice for implementing acoustic echo cancellation system. Additionally, it does only require 2NX 1 multiplication operations.



Fig. 1. Output of FIR Filter



Figure 2. The time diagram of implementing an 8-bit /16-bit input/output 4 weight adaptive filter.



Figure 3. Top level Simulation Results for LMS

References

- [1] B. Widrow and S.D.Stearns," Adaptive Signal Processing", Prentice-Hall, Englewood Cliffs, N.J., 1985.
- [2] S. Haykin, "Adaptive Filter Theory", Fourth Edition, Prentice Hall, Upper Saddle River, N.J., 2002.

- [3] Reid M. Hewlitt, "Canonical Signed Digit Representation for Fir Digital Filters", IEEE Workshop on Signal Processing Systems, 2000,pp.416-426.
- [4] Vijender Saini, Balwinder Singh, and Rekha Devi "Area Optimization of FIR Filter and its Implementation on FPGA" International Journal of Recent Trends in Engineering, Vol 1, No. 4, May 2009
- [5] Petr Sysel, Ondřej Krajsa "Optimization of FIR filter implementation for FMT on VLIW DSP" ISSN: 1792-4324.
- [6] Levent Aksoy, Ece Olcay Gunes "Area Optimization Algorithms in High-Speed Digital FIR Filter Synthesis" SBCCI'08, September 1–4, 2008, Gramado, Brazil.
- [7] Shanthala S, S. Y. Kulkarni "European Journal of Scientific Research, ISSN 1450-216X Vol.31 No.1 (2009), pp. 19-28
- [8] Roger Woods, John McAllister, Ying Yi "FPGA-based Implementation of Signal Processing Systems" 2008 John Wiley & Sons, Ltd, 2008
- [9] Ahmed Elhossini, Shawki Areibi, Robert Dony, "An FPGA Implementation of the LMS Adaptive Filter for Audio Processing", IEEE International Conference on Reconfigurable Computing and FPGA's, ReConFig 2006.

International Journal of Computer and Communication Technology (IJCCT), ISSN: 2231-0371, Vol-3, Iss-4