International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869, Volume-3, Issue-6, June 2015

Hardware Implementation of Adaptive System Identification

Kusha Tyagi

***Abstract*— In this paper, we present a semicustom design for a FPGA implementation of a Adaptive filter. We refer delayed least means square (DLMS) adaptive algorithm to accomplish lower adaptation-delay. We have use single precision floating point (IEEE 754 standard) data format for all inner-product computation.**

Besides, we have proposed an efficient pipelined architecture for the implementation of this adaptive filter. We have shown that the proposed DLMS adaptive filter can be implemented by a pipelined inner-product computation unit for calculation of feedback error, and a pipelined weight-update unit for filter order N.

***Index Terms*— FIR, LMS, Floating point.**

I. INTRODUCTION

Adaptive digital filters are extensively used in many DSP applications, such as, noise and echo abolition, system identification, and channel equalization etc. [1]. Most of these applications involve real-time adaptive filtering, and therefore, they require to be realized through dedicated VLSI systems. The LMS-based FIR adaptive filter is the most trendy one due to its inherent simplicity and satisfactory convergence performance [2]. The straight-forward implementation of LMS adaptive filter involves a large critical-path. Therefore, for adaptive filtering of input signal with high sampling-frequency, it is necessary to decrease the critical-path by pipelined implementation. Since the conventional LMS algorithm does not support pipelined execution due its recursive behavior, it is modified to a form called delayed LMS algorithm [3], which supports pipelined execution. A lot of work has been done to execute the delayed LMS algorithm to increase the maximum usable frequency.

In this paper, we propose a VHDL implementation of modified DLMS algorithm and a new architecture for high-speed adaptive filtering with very low adaptation-delay. The proposed architecture involves three pipelined blocks: (i) Basic FIR circuit which may cause noise, (ii) one for the calculating error, (iii) and the other for weight-redefine. The subtraction of FIR and adaptive circuit generate error which is minimized by weight updating, and realized by a pipelined inner-product unit.

The multiplications and additions involved in weight updating are single precision floating point multiplier and adder. It is set up to be more efficient in terms of power-delay product and area-delay product compared with the existing design, besides, it is easily scalable for higher order filters.

Kusha Tyagi, M.Tech. Student, MNIT, Jaipur, India.

II. THE DELAYED LMS ALGORITHM

For each input sample, the algorithm computes the FIR filter output considering noise in it, calculate the error by difference between the computed output and the desired response , and makes use of that difference to update the filter weights in each cycle. During the n-th iteration, LMS algorithm updates the weights as follows:

$$W(n+1) = w(n) + \mu \cdot e(n) \cdot x(n) \quad (1)$$

where

$$e(n) = y(n) - yeff(n) \quad (2)$$

$$y(n) = h^T(n)*x(n)$$

$$yeff(n) = w^T(n) *x(n) \quad (3)$$

x(n) is the input vector for basic FIR, h(n) is coefficient and w(n) is the weight vector of an Nth order LMS adaptive filter at the nth iteration, respectively, y(n) is the desired response and yeff(n) is the adaptive filter output of the nth iteration. e(n) is the error in the nth iteration which is used to update the weights, and μ is the convergence-factor(0.15).

The DLMS algorithm, instead of using the recent-most feedback-error e(n) equivalent to the n-th iteration for updating the filter weights, it uses the delayed error e(n), i.e. the error equivalent to (n)-th iteration for updating the current weight. The weight-update equation of DLMS algorithm is given by

$$W(n+1) = w(n) + \mu.e(n).x(n) \quad (4)$$

where m is the adaptation-delay. The structure of conventional delayed LMS adaptive filter is shown in Fig. 1. It can be seen that the adaptation-delay m is the number of cycles required for the error corresponding to any given sampling instant to become available to the weight adaptation circuit.

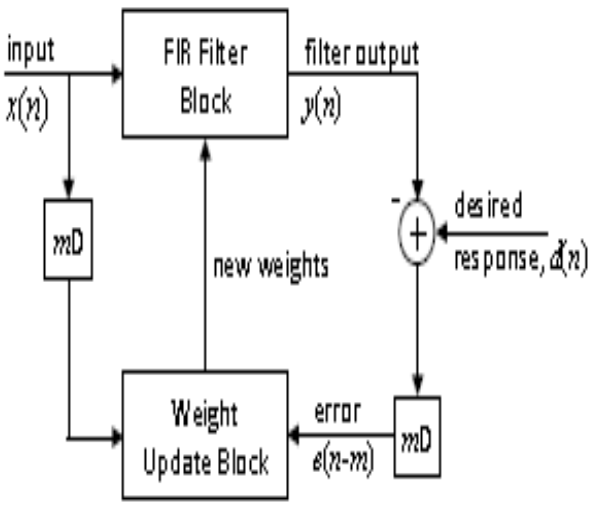


Fig. 1 Structure of conventional delayed LMS adaptive filter.

III. PROPOSED DESIGN

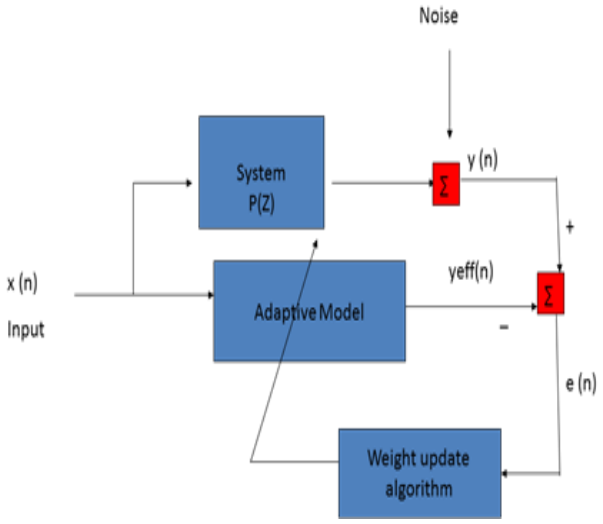


Fig. 2 Proposed design of adaptive filter

System Identification is a task of identifying an existing unknown system by an appropriate model. The unknown system is FIR filter, the appropriated model is adaptive filter. The basic operation involves three processes

1. **Filtering process** produces an output signal in response to a given input signal.
2. **Adaptation process** aims to adjust the filter parameters (filter transfer function) to the (possibly time-varying) environment.
3. **Weight update process** to calculate require coefficient for minimizing the error.

IV. IMPLEMENTATION OF ADAPTIVE FILTER USING LMS ALGORITHM IN MATLAB

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 .It uses Mean Square Error (MSE) as a criterion. LMS uses a step size parameter, input signal and the difference of desired signal and filter output signal to frequently calculate the update of the filter coefficients set.

The choice of the step-size parameter and the order of the filter effectively determine the performance of LMS. 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. 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. In order to select appropriate step size and filter order, MATLAB simulation of LMS algorithm is carried out. Based on the simulation results the adaptive parameters obtained will be applied to the hardware implementation process of LMS algorithm.

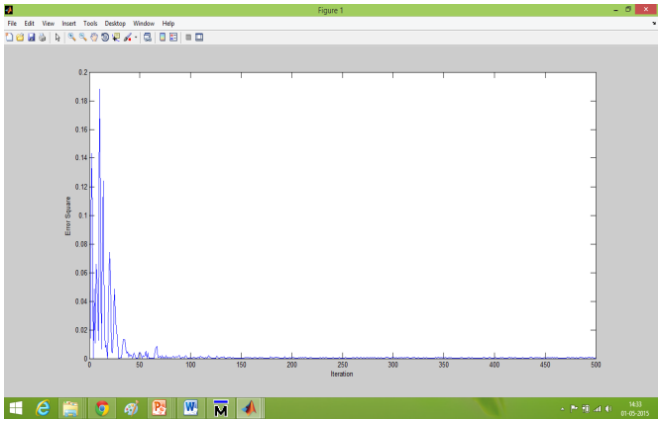


Figure 3: Error square as of function of iteration

The error is calculated for random generation of 500 inputs with the help of the equation error=y-yeff where y is the output of the fir filter and yeff is the output of adaptive filter.

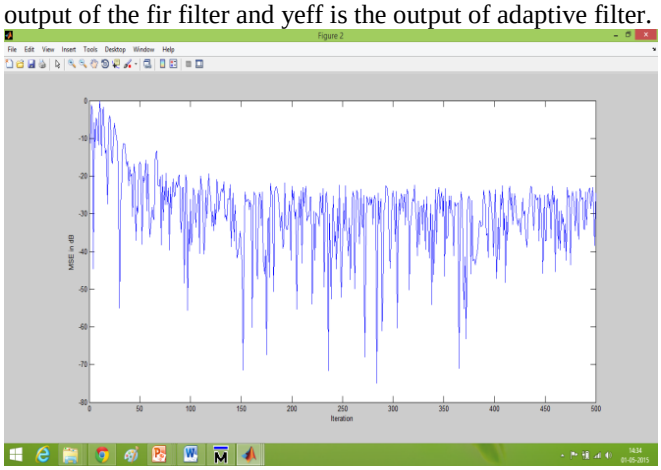


Figure 4 : Mean Square Error

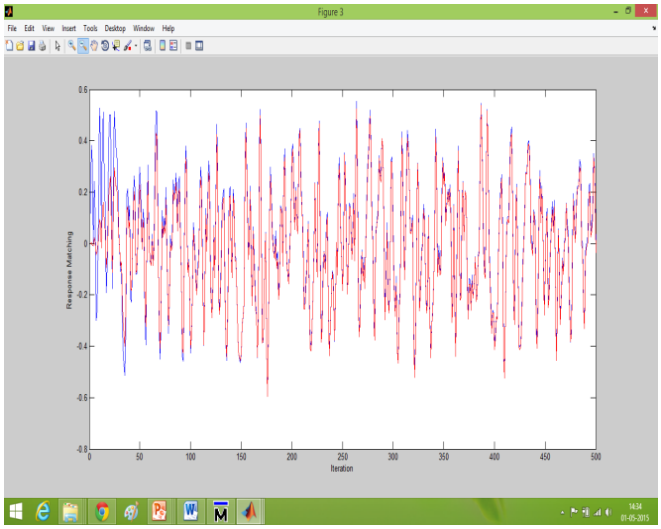


Figure 5 : Response matching

V. CHALLENGES OF HARDWARE IMPLEMENTATION

- In matlab implementation we found all coefficient, input and output are real number and in vhdl real number is not synthesisable.
- Data of 500 input should be same as MatLab for comparision.

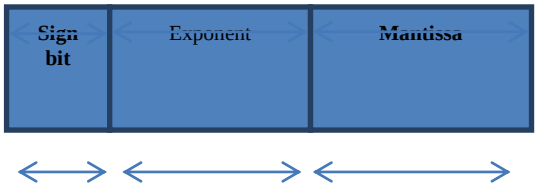
Approach

To overcome above mention issue we have used single precision 32 bit floating point representation for all data, input and output. And for synchronizing inputs with matlab we have used file handling text io commands of vhdl.

VI. SINGLE PRECISION (32 BIT) FLOATING POINT (IEEE754)

The single-precision number is 32 bit wide. The single precision number has three main fields that are sign, exponent and mantissa. The 24-bit mantissa (the leading one is implicit) can approximately represent a 7-digit decimal number, while an 8-bit exponent to an implied base of 2 provides a scale factor with a reasonable range. Thus, a total of 32 Bit is needed for single-precision number representation. To achieve a bias equal to $2^{n-1} - 1$ is added to the actual exponent in order to obtain the stored exponent. This equals 127 for an eight-bit exponent of the single-precision format. The addition of bias allows the use of an exponent in the range from −127 to +128, corresponding to a range of 0-255 for singleprecision number. The single-precision format offers a range from 2^{-127} to 2^{+127}, which is equivalent to 10^{-38} to 10^{+38}

Sign: 1-bit wide and used to denote the sign of the number i.e. 0 indicate positive number and 1 represent negative number.
Exponent: 8-bit wide and signed exponent in excess-127 representation.
Mantissa: 23-bit wide and fractional component



VII. VHDL IMPLEMENTATION OF SYSTEM IDENTIFICATION

We have generated 500 random inputs from Matlab (discrete value) , the real numbers both positive & negative for input & 500 for noise. We have changed the 500 inputs from real to floating point numbers with the help of Matlab code. These 500 floating point inputs and noise value are saved in text file. With the help of File handling inputs from text file are inserted into separate rom for input and noise.
The inputs from rom are getting inserted into FIR filter as well as to the adaptive filter.we will get the output from fir filter named y.
To assume system noise we add the contents of noise stored in rom to y and store the results in another rom.
The same inputs in rom are also going to adaptive filter .The output of adaptive filter is yeff. So error is generated as e=y-yeff
This error is used to update the weights of adaptive filter with the help of the LMS equation.
$W(n+1) = w(n) + \mu \cdot e(n) \cdot x(n)$

This LMS algorithm will update the weights of adaptive filter in such a way so that the output of both adaptive filter as well as fir filter will almost be same and error will be minimized.

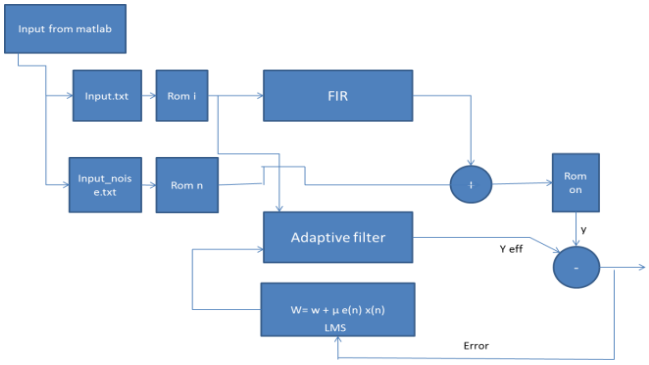


Figure 7: Designed system of Implementation

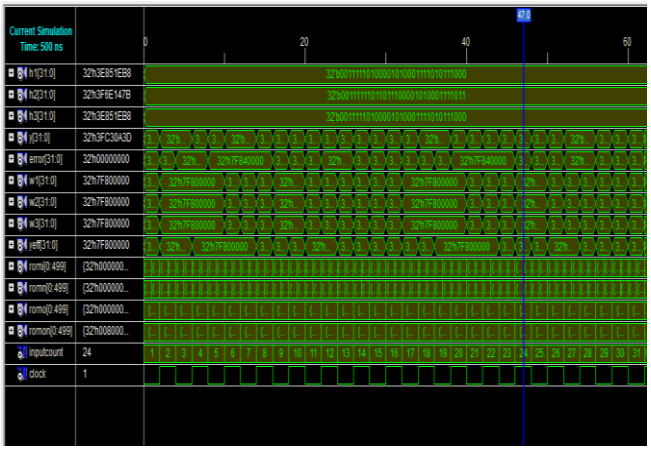


Figure 8: Simulation Result of Xilinx

VIII. ERROR COMPARISON (INITIAL 10 ERRORS)

	Input	Error in MatLab	Error in VHDL
1	0.3587	-0.0881	-0.0721
2	0.4205	-0.4304	-0.1983
3	0.2508	-0.3247	-0.2985
4	-0.2144	-0.0528	-0.0510
5	0.2968	0.1730	0.0987
6	-0.3572	0.3665	0.2246
7	0.0046	0.0547	0.0313
8	0.1107	0.0118	0.0014
9	0.2038	0.1586	0.0812
10	-0.1167	0.0635	0.0417

Table 1- Comparison of MatLab and VHDL error(for first 10 inputs)

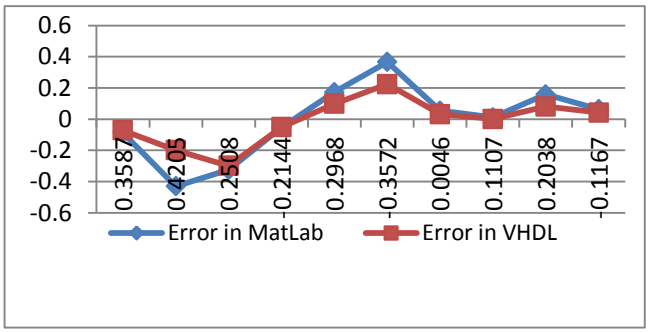


Fig 9 – Graphical representation of comparison of error

IX. Conclusion

The MatLab implementation of System identification Adaptive Filter is carried out along with VHDL implementation. This system is capable to identify coefficient and minimize the effect of noise and error, and provides desire output. The coefficient recalculated by LMS algorithm which is quite faster than other algorithms.

Reference

[1] SYSTEM IDENTIFICATION USING ADAPTIVE FILTER ALGORITHMS, IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) ISSN: 2278-2834-, ISBN: 2278-8735, PP: 54-59

[2] ”SYSTEM IDENTIFICATION WITH LEAST MEAN SQUARE ADAPTIVE ALGORITHM” INTERDISCIPLINARITY IN ENGINEERING SCIENTIFIC INTERNATIONAL CONFERENCE, TG. MUREŞ – ROMÂNIA, 15 -16 November 2007

[3] A High-Speed FIR Adaptive Filter Architecture using a Modified Delayed LMS Algorithm-4244-9474-3/11/ ©2011 IEEE

[4] SIMULATION AND SYNTHESIS OF 32-BIT MULTIPLIER USING vhdlinternational Journal of Advances in Engineering & Technology, Jan. 2013. ©IJAET ISSN: 2231-1963

[5] Design and implementation of efficient 32-bit floating point multiplier using Verilog International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume 2 Issue 6 June 2013 Page No. 2098-210

Kusha Tyagi is doing M.Tech. from MNIT, Jaipur.