VLSI Implementation of Adaptive Filter On Reconfigurable Platform

M.S. Sutaoane , M.B.Mali and Sunita R. Deo

Abstract - Digital filtering algorithms are most commonly implemented using general purpose digital signal processing chips or special purpose digital filtering chips and application-specific integrated circuits (ASICs) for higher rates. Filtering data in real-time requires dedicated hardware to meet demanding time requirements. If the statistics of the signal are not known, then adaptive filtering algorithms can be implemented to estimate the signals statistics iteratively. Modern field programmable gate arrays (FPGAs) include the resources needed to design efficient filtering structures.

This paper describes an approach to the implementation of digital filter algorithms based on reconfigurable platform i.e. field programmable gate arrays (FPGAs). Advancements in Field Programmable Gate Arrays provide new options for DSP design engineers. The FPGA maintains the advantages of custom functionality like an ASIC while avoiding the high development costs and the inability to make design modifications after production. The FPGA also adds design flexibility and adaptability with optimal device utilization while conserving both board space and system power, which is often not the case with DSP chips. When a design demands the use of a DSP, or time to market is critical, design adaptability is crucial, and then FPGA may offer a better solution.

***Index Terms*--Adaptive filters, Adaptive signal processing, Digital filters, Digital signal processors, Field programmable gate arrays, Filtering**

I. INTRODUCTION

There are many advantages to hardware that can be reconfigured with different programming files. Dedicated hardware can provide the highest processing performance, but is infl[1]exible for changes. Reconfigurable hardware devices offer both the flexibility of computer software, and the ability to construct custom high performance computing circuits. The hardware can swap out configurations based on the task at hand, effectively multiplying the amount of physical hardware available.

The most obvious problem in custom VLSI approach is the lack of flexibility. Custom devices are often suited only for use in a particular application, and cannot be easily reconfigured for other operations even within that same domain. Another problem, which the custom VLSI approach often imposes, is a lack of adaptability once a device is in use within a system. Although some problems can be overcome with sufficient forethought, the costs in performance, implementation complexity, and additional design time often preclude flexible solutions.

Lack of flexibility can forestall the cost-effective evaluation of exotic algorithms in a high performance real-time environment. Only high volume applications or extremely critical low volume applications can justify the expense of developing a full custom solution. There are a variety of algorithms which are not within the performance

envelope of general purpose processors, and which are not sufficiently commonplace or well-understood to justify implementation in a full custom design. These algorithms cannot be evaluated with the traditional approaches.

The advantages of the FPGA approach to digital filter implementation include higher sampling rates than are available from traditional DSP chips, lower costs than an ASIC for moderate volume applications, and more flexibility than the alternate approaches.

II. BACKGROUND

A non-adaptive filter has predefined function i.e. it may be low pass, high pass, band pass or band stop filter. However features such as noise in the environment may affect the filtered result. Adaptive filters on the other hand, may be designed to adjust to their environment such that the filter may adapt to noise so as to produce the desired result. Adaptive filters learn the statistics of their operating environment and continually adjust their parameters accordingly.

In practice, signals of interest often become contaminated by noise or other signals occupying the same band of frequency. When the signal of interest and the noise reside in separate frequency bands, conventional linear filters are able to extract the desired signal [1]. However, when there is spectral overlap between the signal and noise, or the signal or interfering signal's statistics change with time, fixed coefficient filters are inappropriate. Fig. 1 shows an example of a wideband signal whose Fourier spectrum overlaps a narrowband interfering signal.

Prof. Dr. M. S. Sutaoane, is with Government College of Enginnering,Pune-5,Maharashtra,India. (e-mail: mssutaone@gmail.com).

Prof. M. B. Mali,is with Sinhgad College of Engineering,Pune-41, Maharashtra,India. (e-mail: madanmali@gmail.com).

Sunita R. Deo is with Sinhgad College of Engineering,Pune-41, Maharashtra,India. (e-mail: sunit_sagat2002@yahoo.com).

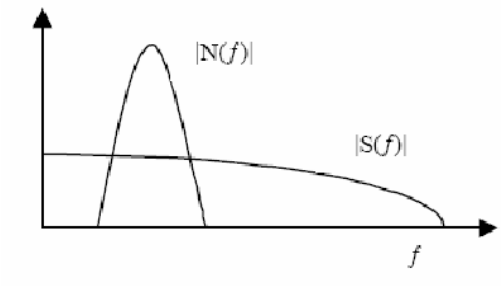

Fig. 1. A strong narrowband interference N(f) in a wideband signal S(f)

This situation can occur frequently when there are various modulation technologies operating in the same range of frequencies. In fact, in mobile radio systems cochannel interference is often the limiting factor rather than thermal or other noise sources. It may also be the result of intentional signal jamming, a scenario that regularly arises in military operations when competing sides intentionally broadcast signals to disrupt their enemies' communications. Furthermore, if the statistics of the noise are not known a priori, or change over time, the coefficients of the filter cannot be specified in advance. In these situations, adaptive algorithms are needed in order to continuously update the filter coefficients.

The desired result is itself given in the filter's specifications. This tells the type of filter and the type of input signals that the filter can handle. However if we wish to a more flexible filter design then the filter type itself should be adjustable. This may be adjusted to cope with different input signals so as to produce the desired output signal or the specification of the filter itself may be changed to change the desired output signal [1].

A. Adaptive filter Algorithm – Least Mean Square (LMS)

The Least Mean Square adaptive algorithm is a simple well-behaved algorithm, which is commonly used in applications where a system has to adapt to its environment.

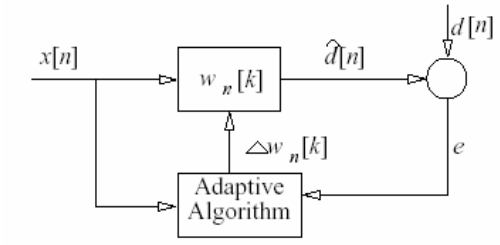

Fig. 2. Block diagram of an adaptive filter

Fig. 2 shows a block diagram of how an adaptive filter can be formulated in an equalizer setting. In this case, the filter *w* is adapting to produce an output sequence $\hat{d}[n]$ which is identical to a known output d[n]. *T*he filter *w* as being of FIR type, with *p* coefficients, i.e.

$$w_n = [w[1]w[2]...w[p]]^T \quad (1)$$

The subscript *n* indicates that the filter coefficients themselves vary with time. The adaptation algorithm calculates the update based on knowledge of the input, and on an error signal e. During training, such a signal can be generated by using a known training sequence at both receiver and transmitter. The Least Mean Square (LMS) Algorithm is the most widely used technique to find an update equation for the system shown in Fig. 2.

For the LMS algorithm, the coefficient vector update equation becomes:

$$w_{n+1} = w_n + \mu e[n]x[n] \quad (2)$$

where

$$e[n] = \hat{d}[n] - d[n] \quad (3)$$

and μ is a scalar vector called step size.

Because of the ability of adaptive filter to perform well in unknown environments and track statistical time variations, adaptive filters have been employed in a wide range of fields However, there are essentially four basic classes of applications for adaptive filters. These are: *Identification*, *inverse modeling*, *prediction*, and *interference cancellation*, with the main difference between them being the manner in which the desired response is extracted. The adjustable parameters that are dependent upon the applications at hand are the number of filter taps, choice of FIR or IIR, choice of training algorithm, and the learning rate [2].

III. FIELD PROGRAMMABLE GATE ARRAY

Traditionally, digital signal processing (DSP) algorithms are implemented using general-purpose (programmable) DSP chips for low-rate applications, or special-purpose (fixed function) DSP chip-sets and application-specific integrated circuits (ASICs) for higher rates.

The SRAM-based FPGA is well suited for arithmetic, including Multiply & Accumulate (MAC) intensive DSP functions. A wide range of arithmetic functions (such as FFT's), convolutions, and other filtering algorithms) can be integrated with surrounding peripheral circuitry. The FPGA can also be reconfigured on the fly to perform one of many system level functions. When building a DSP system in an FPGA, the design can take advantage of parallel structures and arithmetic algorithms to minimize resources and exceed the performance of single or multiple general-purpose DSP devices.

Such a device requires 20nsec per Tap to implement a 16-Tap FIR filter, which translates to a theoretical maximum (with zero wait-states) sample rate of 3.125 million samples per second. An In-System Programmable (ISP) FPGA can also be reconfigured on the board during system operation. Taking advantage of the reconfigurability feature means a minimal chip solution can be transformed to perform multiple functions. For example, an FPGA could be the basis for a system that performs one of several DSP functions. Suppose, for instance, one function is to compress a data stream in transmit mode and another function is to decompress the data

in receive mode. The FPGA can be reconfigured on the fly to switch, or toggle, from one function to another. This capability of the FPGA adds functionality and processing power to a minimum-chip DSP system controlled with an internal or an external controller. The FPGA design cycle requires less hardware-specific knowledge than most DSP chips or ASIC design solutions. Smaller design groups, with less experienced engineers, can design larger, more complex DSP systems in less time than larger design groups with more experienced engineers who are required to know device specific programming languages. The FPGA-based DSP system-level design team can design, test, verify, and ready a complex DSP system for production in weeks [3],[4].

IV. IMPLEMENTATION

The LMS algorithm introduced in previous section is described using the flowchart (refer fig. 3) given below [1].

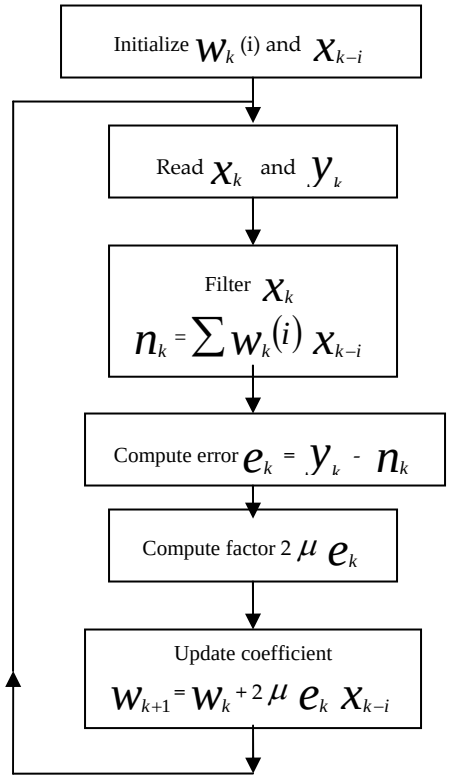

Fig. 3. Flowchart of LMS Algorithm

A. *Hardware Implementation of LMS Adaptive Filter*

Adaptive filter block diagram is presented below. The filter is divided into four main blocks i.e. LMS filter, weight update logic, weight update controller and single tap [9]. The blocks are shown in fig.4,5,6,& 7 resp. For hardware implementation of adaptive filter a VHDL code is written for all these modules. Modelsim simulator is used to simulate the design. Design is synthesized using **Synplify** from Libero IDE.

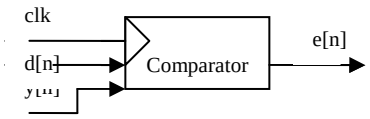

Fig.4. Weight update controller

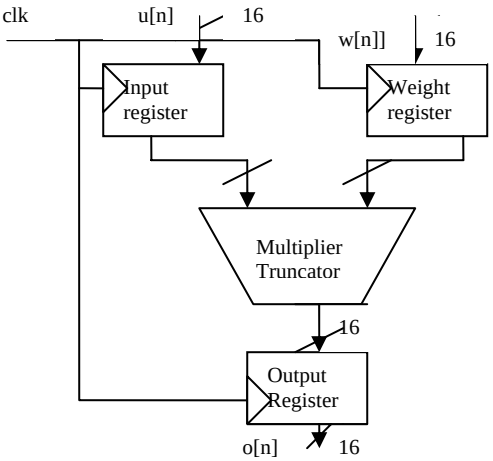

Fig 5. Center Tap

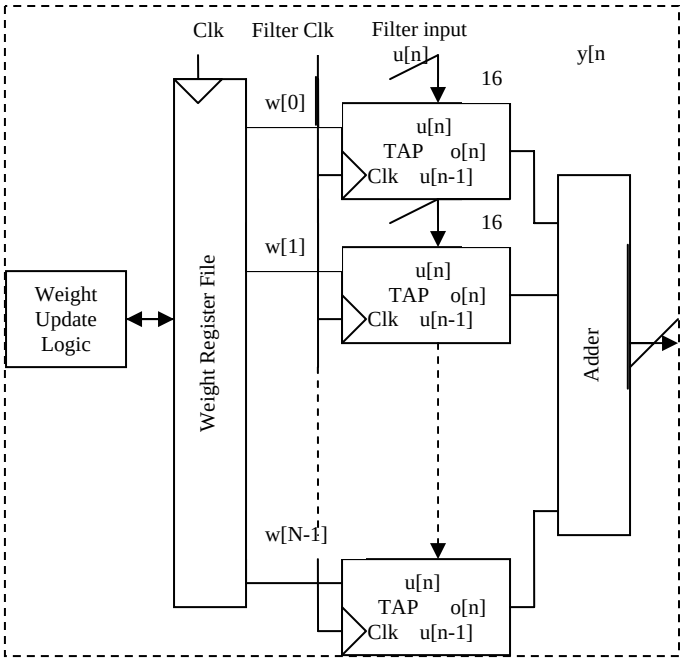

Fig. 6 LMS Filter

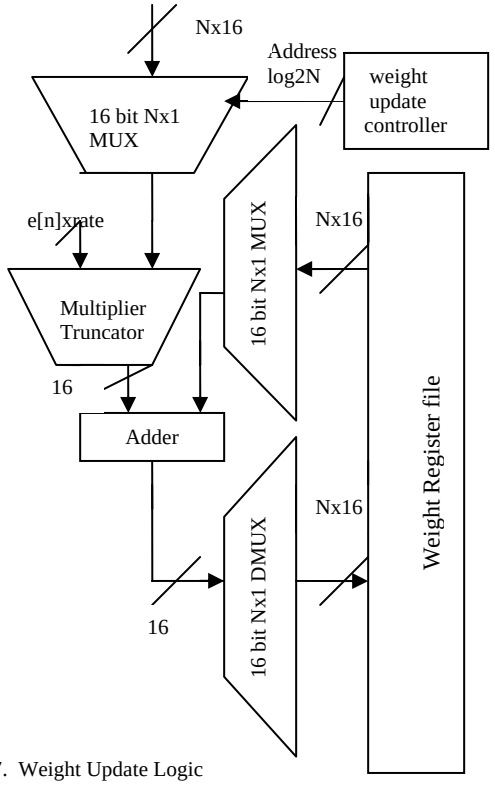

Fig. 7. Weight Update Logic

B. Matlab Simulation of Adaptive filter configured in system identification mode

Here a sinusoidal signal is applied to the adaptive filter as a desired signal i.e. d(n).A corrupted sinusoidal signal is applied as an input signal which is composed of desired signal and a random noise. i.e. x(n) = d(n) + s(n).

Output signal, desired signal and the noisy input signal are plotted and compared in fig. 8.

C. Matlab Simulation of Adaptive filter configured in noise reduction mode

A sinusoidal signal contaminated by noise is applied as an input signal to the adaptive filter. A pure sinusoidal signal is applied as a desired signal to the same. The LMS algorithm iterations are executed till the error signal becomes zero. After successful noise reduction, the spectrum of output signal available at the output of adaptive filter is same as that of the spectrum of desired input sinusoidal signal (refer fig.

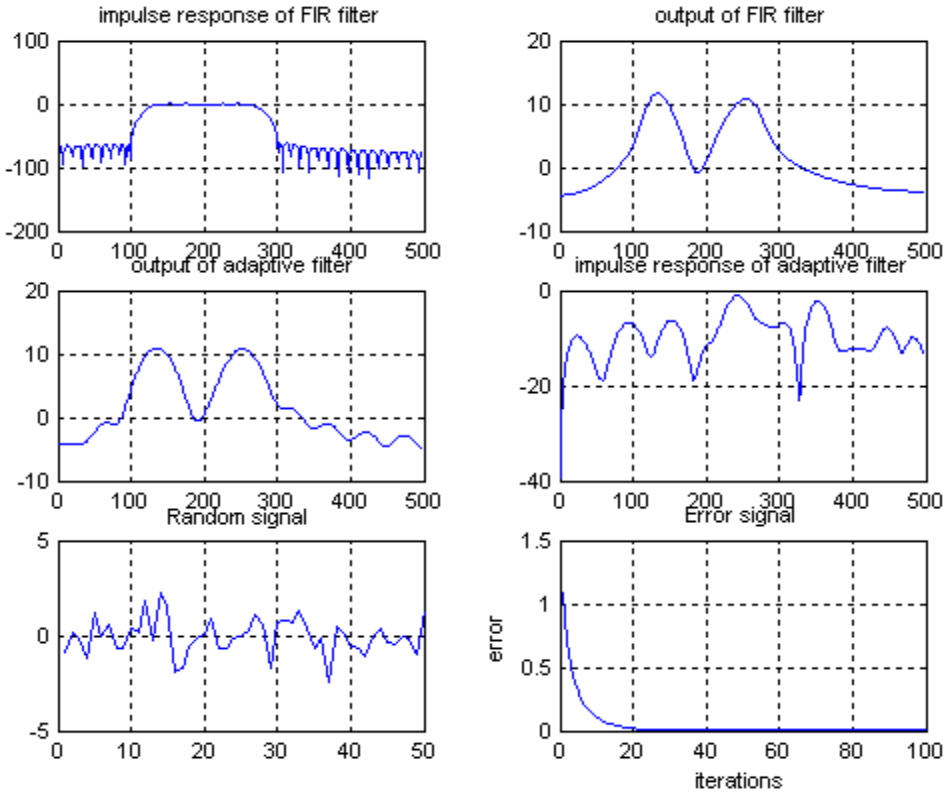

Fig. 8. Matlab Simulation of an adaptive filter configured in system identification mode

V. RESULTS

a) Matlab simulation results of adaptive filter configured in system identification mode are presented below. As can be seen from the plots, the output of adaptive filter is exactly same as that of the FIR filter, which is nothing but the unknown system after forty iterations. The error plot is also presented wherein we can see that the error becomes zero and remains zero after 40 iterations of LMS algorithm. A random signal is applied as an input to adaptive filter as well as unknown system (FIR filter 150Hz-250Hz).

b) Matlab simulation results of adaptive filter configured in noise reduction mode are presented below in fig. 9. A sinusoidal signal contaminated by noise is applied as an input signal to the adaptive filter. A pure sinusoidal signal is applied as a desired signal to the same. The LMS algorithm iterations are executed till the error signal becomes zero. After successful noise reduction, the spectrum of output signal available at the output of adaptive filter is same as that of the spectrum of desired input sinusoidal signal.

c) VHDL code simulation results for noise reduction mode of adaptive filter are obtained in fig.10.The same noisy input signal and desired signal are applied to the VHDL code of adaptive filter as that of signals applied to the MATLAB code. Error obtained in MATLAB simulation is less as compared to error obtained in VHDL simulation.

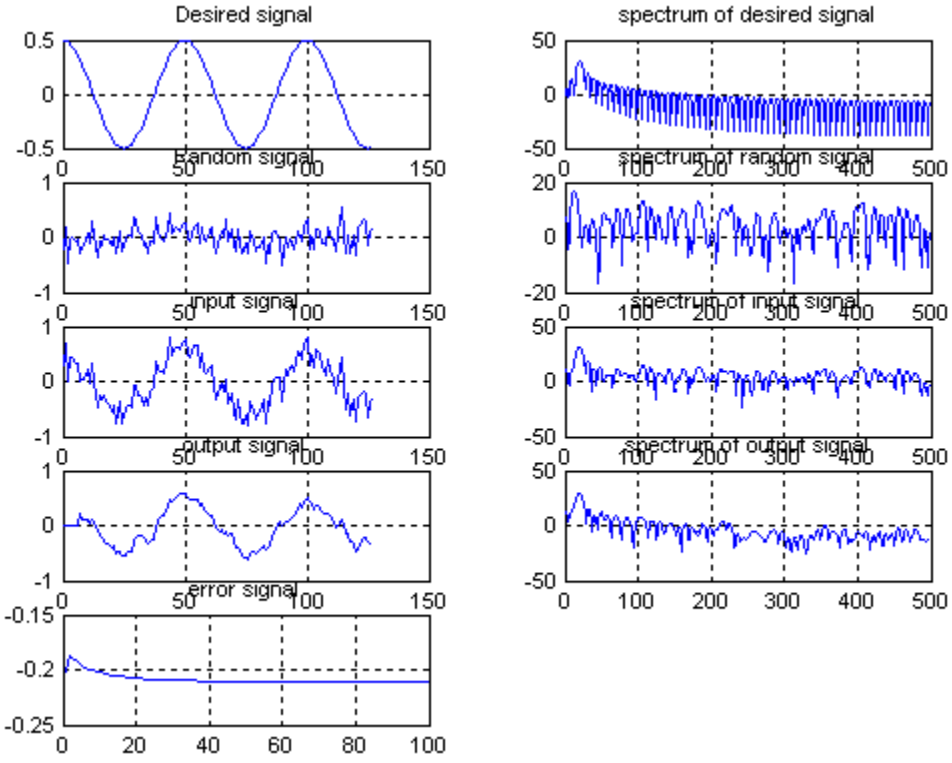

Fig. 9. Matlab Simulation of Adaptive filter in noise reduction mode

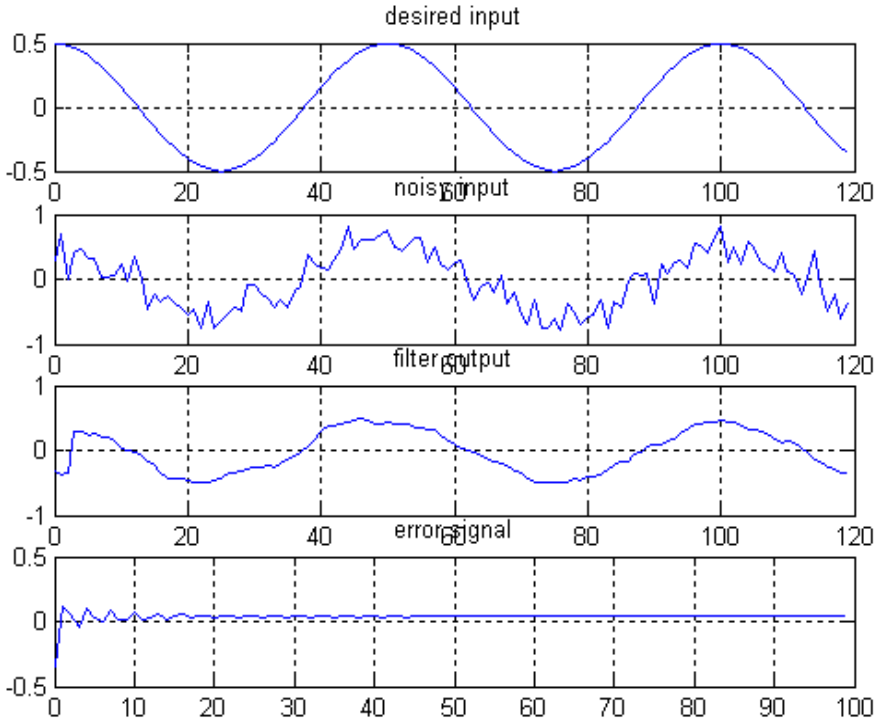

Fig. 10. VHDL code Simulation of Adaptive filter in noise reduction mode

VI. CONCLUSION

This paper has presented the applications and Matlab simulations of adaptive filter. The performance of VHDL code is good but the error reduction obtained by MATLAB code is greater than VHDL code. It has discussed the FPGA implementation approach of adaptive filter. General-purpose DSP implementations often Lack the performance necessary for moderate sampling rates, and ASIC approaches are limited in flexibility and may not be cost effective for many applications. FPGA approach is both flexible and provides performance comparable or superior to traditional approaches.

VII. REFERENCES

[1] E. C. Ifeachor and B. W. Jervis, *Digital Signal Processing, A Practical Approach*, Prentice Hall, 2002.

[2] Haykin Simon, *Adaptive filter theory*, 4th edition, Prentice Hall, New Jersey, 2002.

[3] Lin, A.Y.; Gugel, K.S.; Principe, J.C., "Feasibility of fixed-point transversal adaptive filters in FPGA devices with embedded DSP blocks"
In proceedings of *IEEE International Workshop on System-on-Chip for Real-Time Applications*, 2003.Vol.30 July 2003 Page(s): 157 - 160

[4] Xilinx Inc., "Block Adaptive Filter,"Application Note XAPP 055, Xilinx, CA, January 9,1997.

[5] Proakis, Manolakis, *Digital Signal Processing*, 3rd edition, Prentice Education.

[6] Hardware Description Language." *Wikepedia: The Free Encyclopedia*. 18 May 2004. http://en.wikipedia.org/wiki/Hardware_description_language.

[7] Dimities Manolakis,Vinay K. Ingle,Stephen M. Kogon, *Statistical and Adaptive Signal Processing* ,McGraw-Hill Edition.

[8] Vinay K. Ingle,John G. Proakis, *Digital Signal Processing using MATLAB*, BookWare Companion Series.

[9] Ahmed Elhossini, Shawki Areibi,Robert Dony. "An FPGA Implementation of the LMS Adaptive Filter for Audio Processing" .In proceedings of *IEEE International Conference on Reconfigurable Computing and FPGA's, 2006. ReConFig 2006*. page(s):1-8 Sept.2006

VIII. BIOGRAPHIES

Dr. M. S. Sutaone was born in Nagpur, India on Aug.18,1964. He received his B.E. degree from Nagpur University and M.E. & Ph.D. degree from University of Pune in Electronics and Telecommunication engineering.
He is working as Asst. Professor in Electronics & Telecommunication Engg. Dept. of Government college of Engg., Pune, India. His current research interests are in Signal processing and VLSI architectures.

M.B.Mali was born in Nasik, India on May 25,1968. He received his B.E., M.Tech degree from University of Pune & Department of Electronics, Govt. of India respectively.
He is working as Asst.Prof. in E&TC dept. of SCOE, Pune, India. His current research interests are in Mixed Signal CMOS VLSI Design.

Sunita Deo was born in Pune, India on July 5,1979. She received her B.E. degree from University of Pune & from Electronics & Telecommunication Dept.
She is pursuing her M.E. degree in electronics at University of Pune. Her current research interests are in Adaptive filter implementation on FPGA.