FPGA Implementation of Adaptive LMS Filter

S. S. Godbole[1], P. M. Palsodkar[2] and V.P. Raut[3]

Abstract--**The adaptive filter constitutes an important part of the statistical signal processing. Whenever there is a requirement to process signals that result from operation in an environment of unknown statistics, the use of an adaptive filter offers an attractive solution to the problem as it usually provides a significant improvement in performance over the use of a fixed filter designed by conventional methods. Furthermore, the use of adaptive filters provides new signal-processing capabilities that would not be possible otherwise. We thus find that adaptive filters are successfully applied in such diverse fields as communications, control, radar, sonar, seismology, and biomedical engineering.**

I. Introduction

AN adaptive filter is very generally defined as a filter whose characteristics can be modified to achieve some end or objective and is usually assumed to accomplish this modification or adaptation automatically. It is also usually assumed that the time scale of the modification is very slow compared to the bandwidth of the signal being filtered. With this assumption is that the system designer could in fact use a time-invariant, non-adaptive filter if only the designer knew enough about the input signals to design the filter before its use. This lack of knowledge may spring from true uncertainty about the characteristics of the signal when the filter is turned on, or because the characteristics of the input signal can slowly change during the filters operation. The designer then turns on to an “adaptive” filter, which can “learn” the signal characteristics when first turned on and thereafter can “track” slow changes in these characteristics.

Early adaptive engineering systems have been designed and implemented in the areas of communication and transmission. Nearly three decades, the interest of research in the field of adaptive signal processing has focused on discrete-time. An discrete-time transversal filter ideal tap delay lines has been the dominant form of short-term memory-based filtering for adaptive systems. The reason for its success is, besides the simplicity of the transversal filter structure, error performance, and the existence of fast and efficient adaptive algorithms to adjust its parameters.

In addition, the adaptive filtering process does not require ‘*a priori*’ knowledge of the received signal and operational environment. Coefficient filtering techniques have been used to improve the tracking situation in the adaptive LMS algorithm.

[1] S.S. Godbole is a Lecturer in the Dept. of ECE, G. H. Raisoni College of Engg, Nagpur(e-mail:godbole_123@yahoo.com).

[2] P.M. Palsodkar is a Lecturer, Dept. of ECE, G. H. Raisoni College of Engg, Nagpur

[3] V.P. Raut is a Lecturer, Dept. of ECE, G. H. Raisoni College of Engg, Nagpur

Adaptive filters are widely used in communications to perform such functions as equalization, echo cancellation, noise cancellation, and speech compression. The filter coefficients are determined during a training sequence where a known data pattern is transmitted. The adaptive algorithm adjusts the filter coefficients to force the received data to match the training sequence data. In a modem application, the training sequence occurs after the initial connection is made. After the training sequence is completed, the switches are put in the other position, and the actual data is transmitted. During this time, the error signal is generated by subtracting the input from the output of the adaptive filter.

The input signal is filtered to produce an output that is typically passed on for subsequent processing. This measure of quality, or some function of it , is then passed on to a circuit that uses it to decide how to modify the parameters of the filter to operate until the parameters of the filter are adjusted so that the quality of the filter output is as good as possible. Also, in principle if the characteristics of the input signal or quality assessment change with time, then this assessment/adjustment loop should readjust the filter’s parameters until the new “optimum” output quality is attained.

The functional blocks are quite general and can be chosen in different ways to solve different practical problems. The filter, for example, could be implemented in analog or digital form and could have a tapped-delay-line, pole zero, or lattice structure. The parameters available for adjustment might be the impulse-response sequence values or more complicated functions of the filter’s frequency response. Similarly, the circuit that assesses the quality of the filter output can take several forms, depending on the adaptive filter’s application. The way in which the quality assessment is converted into parameter adjustments, which we term the adaptive algorithm, can also vary.

Adaptive Filter’s study and practical implementation is carried out by using VHDL language. This report describes how exactly the filter is working. The block diagram of Adaptive LMS filter that compromises five inputs as data_in, desired, clk, Reset, train and one output i.e. data_out. Based on the input feeded, when train is equal to 1 then this filter will function as a LMS Adaptive Filter or else simple FIR filter i.e. this train only indicates the differential functioning of two filters. This LMS Adaptive filter is designed using Least Mean Square algorithm. The least mean square algorithm is a linear adaptive filtering algorithm that consist of two basic process:

1) **Filtering Process**, which involves (a) computing the output of a transversal filter produced by a set of tap

inputs, and (b) generating an estimation error by comparing this output to a desired response.

2) **Adaptive Process**, which involves the automatic adjustment of the tap weights of the filter in accordance with the estimation error.

Thus, the combination of these two processes working together constitutes a feedback loop around the LMS algorithm. The transversal filter, around which the LMS algorithm is built: this component is responsible for performing the filtering process. The other one is mechanism for performing the adaptive control process on the tap weights of the transversal filter, hence the designation "adaptive weight control mechanism". This LMS algorithm is developed in QuartusII.

The tap inputs **u (n), u (n-1)…u (n-M+1)** form the elements of the M-by-1 tap input vector **u (n)**, where M-1 is the number of delay elements. Correspondingly, the tap weights **w0 (n), w1 (n)… Wm-1 (n)** form the elements of the M-by-1 tap weight vector **w (n)**.

During the filtering process the desired response d (n) is supplied for processing alongside the tap-input vector u (n). Given this input, the transversal filter produces an output used as an estimate of the desired response d (n). Accordingly, we may define an estimation error e (n) as the difference between the desired response and the actual filter output. The estimation error e (n) and the tap-input vector u (n) are applied to the control mechanism, and the feedback loop around the tap weights is thereby closed. The scaling factor used in this computation is denoted byμ. It is called the step-size parameter. The proposed architecture has been successfully implemented on a FPGA demonstration board for verification.

LMS algorithm is,

$$\mathbf{y(n)} = \sum_{k=0}^{m-1} \mathbf{w^*_k\, u(n-k)}$$

--Output of the Transversal Filter

2) e(n) = d(n) – y(n) -- Error equation

w (n+1) = w(n) + μ .u(n). e(n) --Updated coefficient equation

From the Result of Simulation, after giving inputs like data_in, desired, clk. At each clk event the new process is performed i.e. y(n) the output of FIR is calculated then error and coefficients are updated if necessary. If there is a need then only the filter is updating the coefficients. Similarly by computing the LMS equations, the same result is obtained.

A significant feature of the LMS algorithm is its simplicity. Moreover, it does not require measurements of the pertinent correlation functions, nor does it require matrix inversion. Indeed, it is the simplicity of the LMS algorithm that has made it the standard against which other adaptive filtering algorithms are benchmarked. Thus The LMS algorithm is relatively simple to implement.

In many applications requiring filtering, the necessary frequency response may not be known beforehand, or it may vary with time. In such applications, an adaptive filter which can automatically design itself and which can track system variations in time is extremely useful. Thus Adaptive filters are used extensively in a wide variety of applications.

II. Adaptive Filter

A filter is an important component in the communication world. It can eliminate unwanted signals from useful information. However, to obtain an optimal filtering performance, it requires '*a priori*' knowledge of both the signal and its embedded noise statistical information. The classical approach to this problem is to design frequency selective filters, which approximate the frequency band of the signal of interest and reject those signals outside this frequency band.

The removal of unwanted signals through the use of optimization theory is becoming popular, particularly in the area of adaptive filtering. These filters minimize the mean square of the error signal, which is the difference between the reference signal and the estimated filter output, by removing unwanted signals according to statistical parameters.

Fig.1 shows the Adaptive filter system.

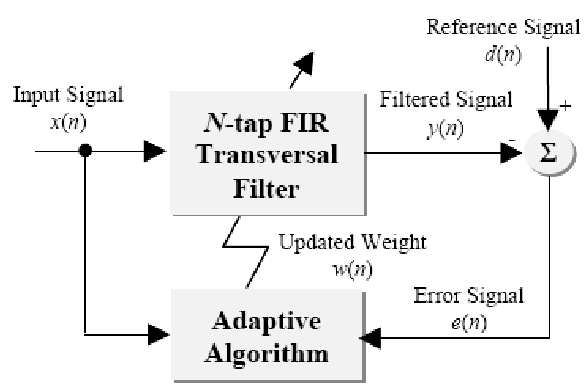

Fig.1: Adaptive filter system

In many filter requiring applications, the necessary frequency response may not be known beforehand, or it may vary with time. In such applications, an adaptive filter which can automatically design itself and which can track system variations in time is extremely useful. Adaptive filters are used extensively in a wide variety of applications, particularly in telecommunications.

III. Applications of Adaptive Filters

The self-adjusting character of adaptive filters allows them to operate in an unknown environment and to track time-variations of the input statistics. This unique character makes the adaptive filter a very powerful device for communication signal processing applications. In these communication systems, adaptive filters are mainly employed in channel equalization, echo cancellation and noise cancellation. In channel equalization, the inter-symbol interference and noise within a transmission channel are removed by using an adaptive filter, which dynamically models the inverse characteristics of the contamination within the channel. For echo cancellation, adaptive filters are able to synthesis the echo signal which contains echoes caused by impedance mismatch in a telephone cable, and then subtract it from the original received signal, thereby removing the echo [10]. In prediction, adaptive filters recreate a narrowband for the primary signal frequency, thus rejecting other unwanted signals or noise.

Adaptive filters have been widely used in radar systems, such as adaptive beam forming and other detection applications in radar signal processing [10] where the receiving antennae have some '*a priori*' information regarding the received radar signals. By further exploiting adaptive filtering techniques, the interference can be reduced. However, in ESM receiver systems, '*a priori*' information is completely unknown. In such cases, an adaptive prediction technique can be used to constitute a narrowband of the received radar signal from the wideband communication channels to reduce the background noise, thus, improving SNR of the received radar pulse. Because of their ability 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 [9] 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.

IV. DESIGN AND SIMULATION

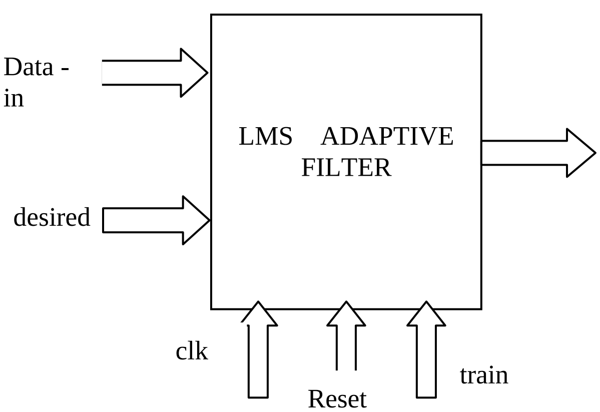

Fig. 2 System Block Diagram

The Figure.6 illustrates the general block diagram of LMS Adaptive Filter that compromises five inputs as data_in, desired, clk, reset, train and one output i.e. data_out. Based on the input feeded, when train = 1 then this filter will function as a LMS Adaptive filter or else simple FIR filter i.e. this train only indicates the differential functioning of two filters.

The figure above shows the detailed structure of the Adaptive Transversal Filter. This structure of Adaptive Transversal Filter is designed in VHDL code and simulated using QUARTUSII tool.

IV. IMPLEMENTATION

A. Device and Tool information:

The design has been implemented in VHDL using QUARTUS II 4.2 tool and its FLEX10KE device family FPGA's. Currently, the whole project one ALTERA FLEX10KE device. FLEX8000, MAX9000, and FLEX10KB ALTERA device families do not support more than 256 words of memory and this design could not fit on them. The behavioral VHDL description of this design is placed and routed on a FLEX10KE device by the QUARTUS II tool.

V. SIMULATION RESULTS

The code is written in a hardware description language called VHDL and it is simulated in QUARTUS II tool. The simulation results at 15ns, 165ns and 305ns obtained practically are found to be equivalent to the results obtained by mathematical calculations.

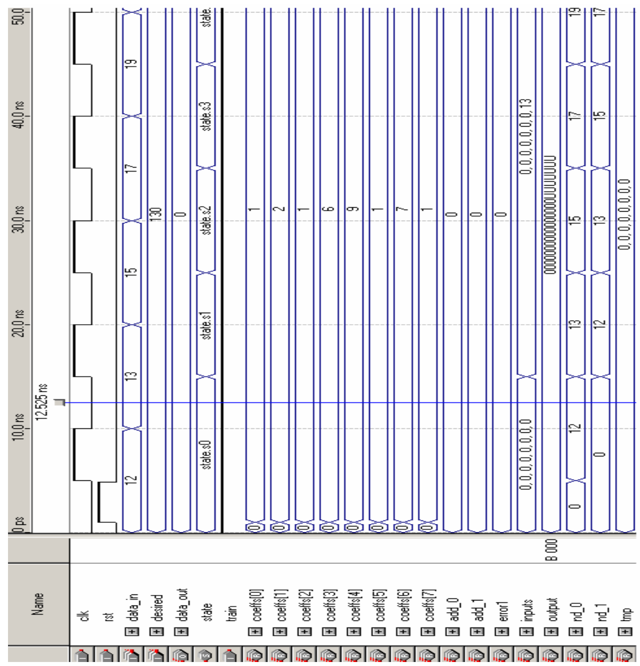

Figure 3 : Simulation Result(at 15ns)

From this result of simulation, it is observed that at **15ns:**

data_in = [13, 0, 0, 0, 0, 0, 0, 0]
coe = [1, 2, 1, 6, 9, 1, 7, 1]
output = 13
data_out = 0

Similarly by computing through the following equations, the result obtained as,

$$\mathbf{y(n)=1/127\ \ Y(n) = \sum_{i=0}^{m-1} w^*_k\, x(n-i)}$$

y(n) =1/127(w0x7 + w1x6 + w2x5 + w3x4 + w2x3 + w1x2 + w0x1)
= (1/127)(1*13) =13/127= 0.1
e(n) = desired – y(n) = 0
w (n+1) = w (n) + u .e(n).x(n) = 0

On comparison of simulation results obtained at 15ns and the results derived from the LMS algorithm are similar.

VI. CONCLUSION

This Thesis uses a hardware description language called VHDL to implement a Adaptive LMS Predictor. The contribution of this Thesis is VHDL implementation of the LMS algorithm and design & VHDL implementation of Adaptive LMS filter.

In this Thesis, strategies & implementation of high-speed FPGA implementation of the LMS filter based on the LMS algorithm is described. The Least Mean-Square algorithm was found to be the most efficient training algorithm for FPGA based adaptive filters.

In terms of the high-speed architecture, the direct-form approach is preferred for design. The architecture based on the LMS algorithm is successfully implemented on FPGA.

This also contains the characteristics of LMS-based algorithms, the FPGA family, and the design methodology in the research. The overall aim is to choose an algorithm that will offer good tracking ability, particularly where the signals are time-varying. A review of adaptive filters shows that the LMS algorithm is still a popular choice for its stable performance and high-speed capability. The processing speed of the algorithm can be further improved by adopting its pipelined versions FPGAs provide a perfect solution for a rapidly changing technology world. Fast time-to market, low cost for small production volume and reprogram ability make FPGA devices an ideal solution for military and university research. 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 QUARTUS II for training as well as the with the LMS algorithm designed in VHDL only.

This design implementation entailed the employment of QUARTUS II software tool from ALTERA. Upon testing the functionality of the filter through simulations, satisfactory functional and timing performances were observed. This design is fit on a single FLEX10KE ALTERA chip.

Implementing the design on an ALTERA FLEX10KE chip and hardware testing and verification of the filter can be done. Finally, simulation and synthesis with Synopsis tools, gate level design, layout and manufacturing of the filter chip could be obtained.

VII. FUTURE SCOPE

In this thesis, the adaptive LMS filter has been successfully implemented on the FPGA. However, such a high-speed adaptive LMS filter architecture can be pipelined, which adversely affects its adaptive filtering performance due to an increased latency in the closed-loop structure. The research therefore concentrates on the fine-grain pipelined architecture with reduced output latency. The thesis can further be taken up for the flexibility of a pipelined design that can be implemented on the FPGA and the LMS FPGA-based architectures. In addition, as research on finer grained pipelining of the LMS algorithm is limited and front-end digital receivers have become realistic in high-speed communication systems, there is a need to improve the high-speed adaptive filter performance.

[1] K.A. Vinger, J. Torresen, “Implementing evolution of FIR-filters efficiently in an FPGA.” Proceeding,. NASA/DoD Conference on Evolvable Hardware, 9-11 July 2003. Pages: 26 – 29

[2] E. C. Ifeachor and B. W. Jervis, Digital Signal Procesing, A Practical Approach, Prentice Hall, 2002.

[3] S.J. Visser, A.S. Dawood, J.A. Williams, “FPGA based real-time adaptive Filtering for space applications,” Proceedings, IEEE International Conference on Field-Programmable Technology, 16-18 Dec. 2002. Pages: 322–326.

[4] K. K. Parhi, VLSI Digital Signal Processing Systems Design and Implementation. New York: Wiley, 1999.

[5] L. –K. Ting, “Algorithms and FPGA implementations of adaptive LMS-based predictors for radar pulse identification. : Ph.D. dissertation. Queen’s Univ. Belfast, N. Ireland, Jul, 2001.

[6] G. Long, F. Ling, and J. G. Proakis, “The LMS algorithm with delayed coefficient adaptation,” IEEE Trans. Acoust., Speech, Signal Process., vol. 37, pp. 1397-1405, Sep. 1989.

[7] T. Kimijima. K. Nishikawa, and H. Kiya, “A pipelined architecture for DLMS algorithm considering both hardware complexity and output latency,” in Proc. IX-Eur. Signal Processing Conf., Sep. 1998, pp.503-506.

[8] R. W. Stewart, J. J. Soraghan, and T. S. Durrani, “Noncanonical FIR filters and adaptive signal processing,: Electron. Lett., vol. 25, pp. 414-415, Mar. 1989.

[9] D. L. Jones.”Learning characteristics of transpose-form LMS adaptive filters,”IEEE Trans, Circuits Sust. II, Analog Digit. Signal Process, .vol. 39, pp. 745-749, Oct. 1992.

[10] [Hayk’96] S. Haykin, Adaptive filter theory. Third Edition, New Jersey: Prentice-Hall, 1996.

[11] [Parh’99] K.K. Parhi, VLSI digital signal processing systems–Design and implementation. New York: Wiley, 1999.

[12] [Shar’98] A.K. Sharma, Programmable logic handbook, PLDs, CPLDs, & FPGAs. New York: McGraw-Hill, 1998.

[13] Theory and Design of Adaptive Filtes by John R. Treichler, C. Richard Johnson, JR, Michael G. Larimore

[14] G.Long, F. Ling, J. G. Proakis, “The LMS algorithm with delayed coefficient daptation”, IEEE Trans. Acoust., Speech, Signal] B. Widrow and S. D. Stearns, Adaptive Signal rocessing. Englewood Cliffs, NJ: Prentice-Hall, 1985.