International Conference on Recent Trends in Engineering Science and Technology (ICRTEST 2017) ISSN: 2321-8169
Volume: 5 Issue: 1(*Special Issue 21-22 January 2017*) 507 - 512

FPGA Implementation Of LMS Algorithm For Audio Applications

Shailesh M. Sakhare
Assistant Professor, SDCE Seukate,Wardha,(India)
shaileshsakhare2008@gmail.com

Abstract- Adaptive filtering techniques are used in a wide range of applications. Adaptive noise canceller is one of the most interesting applications for adaptive filters, especially for the Least Mean Square (LMS) algorithm, due to its strength and calculus simplicity. Noise problems in signals have gained huge attention due to the need of noise-free output signal in numerous communication systems. The adaptive noise cancellation principle is used to remove an unwanted noise from corrupted signal by subtracting it from the corrupted signal. This paper presents an idea behind the hardware designing of ADC and simulation of ADC controller code used for FPGA Implementation of LMS Algorithm for Audio Applications. Since Xilinx Spartan3 FPGA doesn't have a facility of providing audio input directly designing of ADC is very important.

Keywords- LMS, VHDL, Adaptive, FPGA, ADC.

*****

I. INTRODUCTION

Any communication system consists of a transmitter, channel and a receiver connected together for communication purpose. Typically the channel suffers from two major kinds of problems: Intersymbol interference and Noise. The principle of adaptive noise cancellation is used to remove an unwanted noise from corrupted signal by subtracting it from the corrupted signal. Adaptive noise cancellation [1] is a specific type of noise cancellation which makes the use of noise cancellation by subtracting noise signal from a corrupted signal. An operation is controlled in an adaptive manner for the purpose of improved signal to noise ratio. Fig. 1 below illustrates the basic adaptive noise cancelling concept. It is basically a dual-input, closed loop adaptive control system. Digital signal processing spans a wide variety of application areas which includes speech and image processing, communications, networks and so on. The most commonly used tools for the design of signal processing systems are: Application Specific Integrated Circuit (ASIC), Digital Signal Processors (DSP) and FPGA. DSP is well suited for extremely complex math-intensive tasks, but cannot process high sampling rate applications due to its serial architecture. ASIC can meet all the constraints of digital signal processing, however, it lacks flexibility and requires long design cycle. FPGA can overcome the disadvantages of ASIC and DSP with flexibility, time-to-market, risk-mitigation and lower system costs advantages.

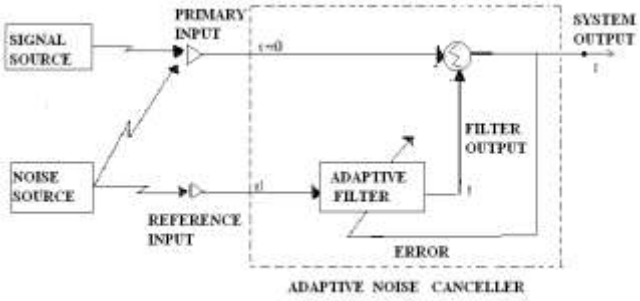

FIG-1: ADAPTIVE NOISE CANCELLING CONCEPT

The concept of adaptive noise cancelling, an alternative method of estimating a signal corrupted by additive noise or interference is to pass it through a filter. The method uses a "primary" input containing the corrupted signal (source + noise) and a "reference" input containing noise correlated in some unknown way with the primary noise. In the ANC system the reference input is processed by an adaptive filter. An adaptive filter differs from a fixed filter in that it automatically adjusts its own impulse response. Adjustment is accomplished through an LMS algorithm that responds to an error signal dependent, among other things, on the filter's output. Thus with the proper algorithm, the filter can operate under changing conditions and can readjust itself continuously to minimize the error signal The error signal used in an adaptive process depends on the nature of the application. The reference input is adaptively filtered and subtracted from the primary input to obtain the signal estimate.

Filtering data in real-time requires dedicated hardware to meet demanding time requirements. The main objective of this paper is to provide an idea behind the designing of

ADC for FPGA Implementation of Least Mean Square Algorithm for Audio Applications, which can be used for Adaptive Noise Canceller. This work will use Xilinx Spartan3 FPGA for synthesis and it doesn't have facility of providing audio input directly. So the first step is to design an ADC for analog to digital conversion. Again whenever we connect any external hardware circuitry to FPGA it must have some controller code so that the communication between external hardware circuit and FPGA takes place easily.

The principal advantage of the method is its adaptive capability, and real time application. The adaptive capability allows the processing of inputs whose properties are unknown. In this research work, hardware implementation of ADC is presented. ADC controller code has been written in VHDL and simulated using Xilinx ISE9.1i & Altera's ModelSim SE 6.3f simulator.

II. LMS ALGORITHM

LMS algorithm, originally proposed by Widrow and others, is widely used for adaptive filter [1], after that; the LMS algorithm with delayed coefficient adaptation was proposed [2]. Delayed LMS (DLMS) algorithm [3] has been derived to achieve low latency. The previous works of Very Large Scale Integrated Circuit (VLSI) implementations of LMS and DLMS are shown in [4]-[6].

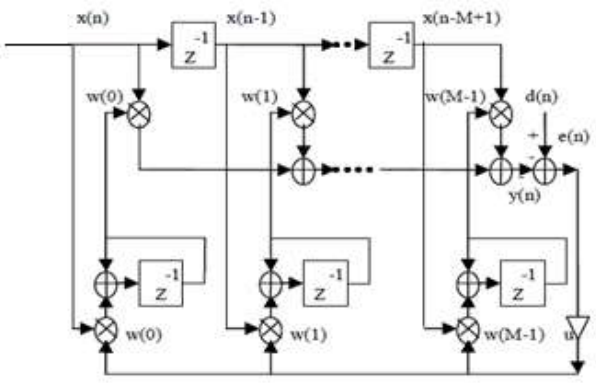

FIG-2: FLOW DIAGRAM OF THE LMS ALGORITHM

However, they are mainly concerned with the convergence behavior of LMS or DLMS, the detailed implemental process of VLSI and the advantage of VLSI implementation are not mentioned clearly. The LMS algorithm is a linear adaptive filtering algorithm which consists of two basic processes.

Filtering Process: This involves (a) computing the output of a transversal filter produced by a set of inputs, and (b) generating an estimation error by comparing this output to a desired response.

Adaptive Process: This involves the automatic adjustment of the tap weights of the filter in accordance with the estimation error.

The LMS algorithm is a widely used algorithm for adaptive filtering. The algorithm is described by the following equations:

$$y(n) = \sum_{i=0}^{M-1} wi(n) * x(n-i); \quad \ldots\ldots\ldots\ldots (1)$$

$$e(n) = d(n) - y(n) \quad \ldots\ldots\ldots\ldots (2)$$

$$wi(n+1) = wi(n) + 2\mu e(n)x(n-i) \quad \ldots\ldots\ldots (3)$$

In these equations, the tap inputs x(n), x(n-1),……,x(n-M+1) form the elements of the reference signal x(n), where M-1 is the number of delay elements. d(n) denotes the primary input signal, e(n) denotes the error signal and constitutes the overall system output. wi(n) denotes the tap weight at the nth iteration. In equation (3), the tap weights update in accordance with the estimation error. And the scaling factor μ is the step-size parameter. μ controls the stability and convergence speed of the LMS algorithm.

III. APPLICATIONS OF LMS ALGORITHM

Noise is an important factor in the operation of any communication system. It is any unwanted signal that corrupts and distorts the desired signal in any way. 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. 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 used for 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 [7], [8].

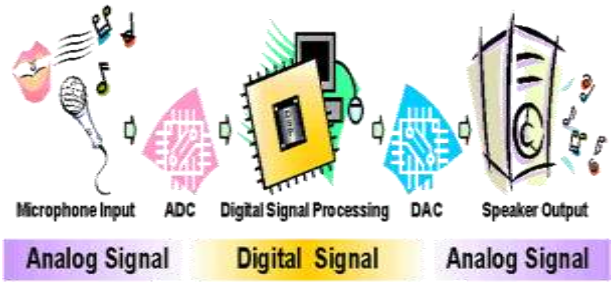

FIG-3: AUDIO SIGNAL PROCESSING

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 [9], [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.

IV. PROPOSED DESIGN

The proposed design approach for the FPGA implementation of Least Mean Square (LMS) algorithm is divided into three phases. Phase-1 is the designing of ADC; to convert audio/analog signal into digital form. Phase-2 FPGA implementation of LMS algorithm and third one is the designing of DAC again to convert the digital signal to original analog form. The proposed block diagram is as shown below.

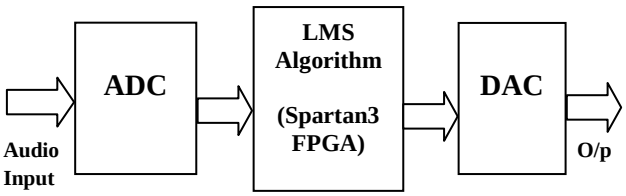

FIG-4: PROPOSED DESIGN FOR FPGA IMPLEMENTATION OF LMS ALGORITHM

V. HARDWARE DESIGNING OF ADC

In electronics, an Analog to Digital Converter (ADC) is a device for converting an analog signal (current, voltage etc.) to a digital code, usually binary. In the real world, most of the signals sensed and processed by humans are analog signals. Analog-to-Digital conversion is the primary means by which analog signal are converted into digital data that can be processed by computers for various purposes [11]. The ADC0808, ADC0809 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8-single-ended analog signals. The device eliminates the need for external zero and full-scale adjustments. Easy interfacing to microprocessors is provided by the latched and decoded multiplexer address inputs and latched TTL TRI-STATE outputs.

Some of the key specifications of ADC0809 are

Resolution	8 Bits
Total Unadjusted Error	±1⁄2 LSB and ±1 LSB
Single Supply	5 V DC
Low Power	15 mW
Conversion Time	100 µs

There are many types of ADC for different applications. The most inexpensive type of ADC is a Successive-Approximation ADC. Inside a Successive-Approximation ADC, a series of digital codes, each corresponds to a fix analog level, are generated successively by an internal counter to compare with the analog signal under conversion. The generation is stopped when the analog level becomes just larger than the analog signal. The digital code corresponds to the analog level is the desired digital representation of the analog signal. The performance of ADCs and DACs mainly depends on their Resolution and Speed. The Resolution of a converter is expressed in the number of Bit. For an ADC, the Resolution states the number of intervals or levels which can be divided from a certain analog input range. An n-bit ADC has the resolution of 1 / 2n. For example, the Resolution of a 16-bit ADC is 1 / 65536, since 216 = 65536. If the measuring voltage range is 10 V, then this input range can be resolved into 10 V / 65536 = 0.153 mV precision.

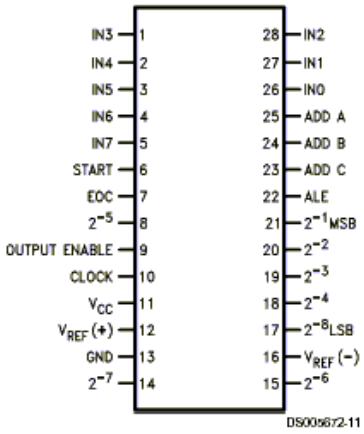

FIG-5.1: CONNECTION DIAGRAM OF ADC0809

The Speed of a converter is expressed by the Sampling Frequency. It is the number of times that the converter samples the analog signal; its unit is Hertz (Hz). In audio signal processing, Sampling Frequencies of 44 kHz, 22 kHz and 11 kHz are mostly used. Using 44 kHz Sampling Frequency means the converter is sampling the analog audio signal and doing analog to digital conversion at 44000 times per second. The higher the Sampling Frequency, the lower the distortion and the better the sound quality.

The components used for hardware designing of ADC are as follows.

1. ADC IC- 0809
2. IC- 555
3. Voltage regulator IC-7812
4. Resistors
5. Capacitors
6. Diodes
7. Microphone

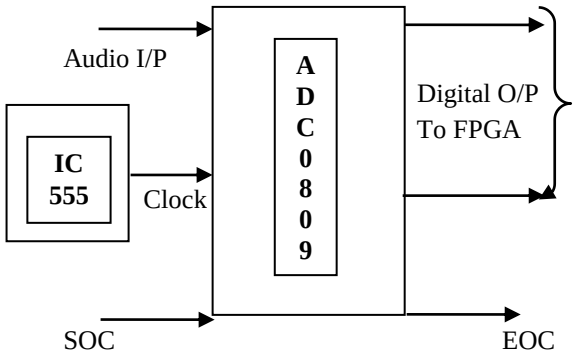

FIG-5.2: BLOCK DIAGRAM OF HARDWARE DESIGNING OF ADC

The circuit of A-to-D converter shown here is configured around ADC 0809, avoiding the use of a microprocessor. The ADC 0809 is an 8-bit A-to-D converter, having data lines D0-D7. It works on the principle of successive approximation. It has a total of eight analogue input channels, out of which any one can be selected using address lines A, B and C. Here, in this case, input channel IN0 is selected by grounding A, B and C address lines. Usually the control signals EOC (end of conversion), SOC (start of conversion), ALE (address latch enable) and OE (output enable) are interfaced by means of a microprocessor. However, the circuit shown here is built to operate in its continuous mode without using any microprocessor. Therefore the input control signals ALE and OE, being active-high, are tied to Vcc (+5 volts).

FIG-5.3: SNAPSHOT OF ADC CIRCUIT (FRONT VIEW)

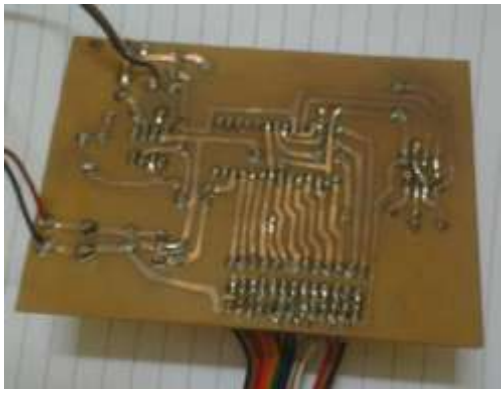

FIG-5.4: SNAPSHOT OF ADC CIRCUIT (BACK VIEW)

As the conversion starts, EOC signal goes high. At next clock pulse EOC output again goes low, and hence SOC is enabled to start the next conversion. Thus, it provides continuous 8-bit digital output corresponding to instantaneous value of analogue input. The maximum level of analogue input voltage should be appropriately scaled down below positive reference (+5V) level. The ADC 0808 IC requires clock signal of typically 550 kHz, which can be easily derived from an astable multivibrator, constructed using 7404 inverter gates. The 8-bit output generated by ADC will be given to Xilinx Spartan-3 (XC3S200ft256) FPGA.

FIG-5.5: FPGA IMPLEMENTATION CIRCUIT

ISSN: 2321-8169
507 - 512

VI. SIMULATION RESULTS

The ADC controller code is written in a hardware description language called VHDL and it is simulated by using XILINX ISE9.1i & ModelSim SE 6.3f simulator tool. The simulation results are observed at various time spans. The code is necessary for Spartan3 FPGA to communicate with ADC so that an audio signal is converted to digital form and it is given as an input to the FPGA.

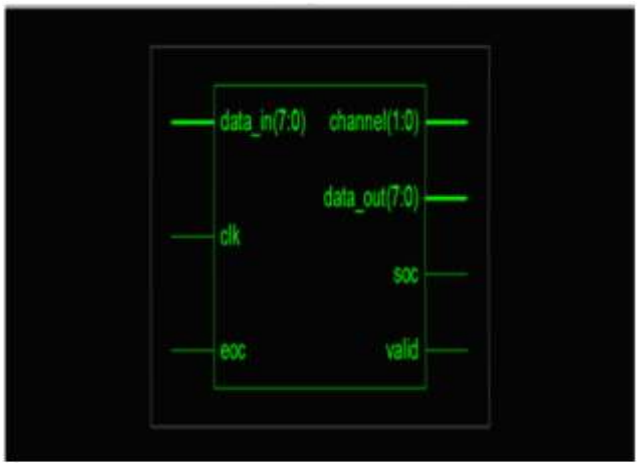

FIG-6.1: RTL SCHEMATIC

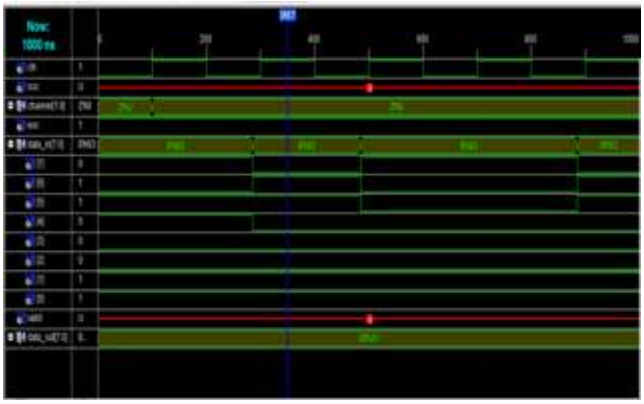

FIG-6.2: SIMULATION RESULTS USING XILINX ISE9.1i SIMULATOR

Since this code is written for ADC to communicate with FPGA that's why the output signals are shown as undefined ('UU'). When we connect the ADC circuit to Xilinx spartan3 FPGA and synthesize the ADC controller code then and only then it will show the output signals. But the alternate method to test the workink of ADC controller code is to simulate the code using ModelSim SE 6.3f simulator. Following are the simulation results for ADC controller using ModelSim SE 6.3f simulator. Fig.6.3 shows the simulation results when SOC is "Low" and EOC is also "Low". As soon as the SOC becomes "High" the conversion begins and EOC becomes active "High" this is shown in fig. 6.4.

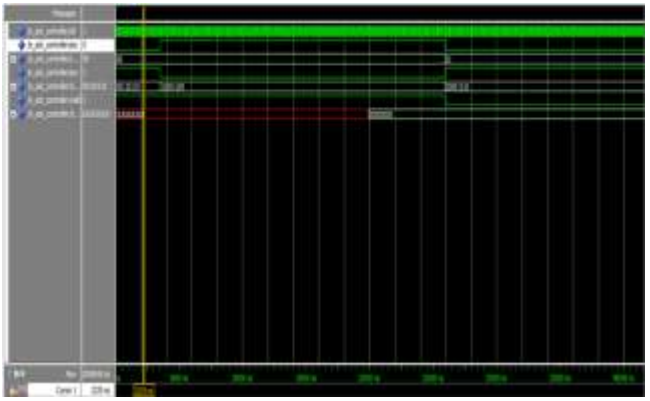

FIG-6.3: SIMULATION RESULTS WHEN SOC= '0' & EOC= '0'

FIG-6.4: SIMULATION RESULTS WHEN SOC= '1' & EOC= '1'

VII. CONCLUSION

This work uses a hardware description language called VHDL for an implementation of Analog to Digital converter required for the FPGA implementation of LMS algorithm for audio applications. The contribution of this research work is VHDL implementation of ADC and hardware designing of ADC required for FPGA implementation of LMS algorithm for audio applications. The Least Mean-Square algorithm was found to be the most efficient training algorithm for FPGA based adaptive filters. The principal advantage of the method is its adaptive capability, and real time application. The adaptive capability allows the processing of inputs whose properties are unknown. The future scope of this work is to implement an adaptive noise canceller using LMS algorithm for audio applications. One of the advantages of this work is that it is real time.

REFERENCES

[1] B. Widrow, J. R. Glover, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, E. Dong and R. C.Goodlin, "Adaptive noise canceling: Principles and

applications", *Proc. IEEE*, vol. 63, Dec. 1975, pp. 1692-1716.

[2] G. Long, F. Ling and J. G. Proakis, "The LMS algorithm with delayed coefficient adaptation," *IEEE Trans. on ASSP*, vol. 37, Sept. 1989, pp. 1397-1405.

[3] C.-L.Wang, "Bit-serial VLSI implementation of delayed LMS adaptive FIR filters," *IEEE Trans. Signal Process.*, vol. 42, Aug. 1994, pp. 2169–2175.

[4] L. K. Ting, R. F. Woods and C. F. N. Cowan, "Virtex FPGA Implementation of a Pipelined Adaptive LMS Predictor for Electronic Support Measures Receivers," *IEEE Trans. VLSI Syst.*, vol. 13, Jan. 2005, pp. 86-95.

[5] M. D. Meyer and D. P. Agrawal, "A high sampling rate delayed LMS filter architecture," *IEEE Trans., Circuits Syst. II, Analog Digit. Signal Process.* vol. 40, Nov. 1993, pp. 727–729.

[6] 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.

[7] 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.

[8] S. F. Boll and D. C. Pulsipher, "Suppression of acoustic noise in speech using two microphone adaptive noise cancellation," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-28, pp. 752–753, 1980.

[9] William Schweber, Electronic Communication System, Prentice-Hall Publication.

[10] Simon Haykin, Adaptive Filter Theory, 4th edition, Prentice Hall, New Jersey, 2002.

[11] www.national.com National Semiconductor Corporation ADC0809 data sheet

[12] www.digilentinc.com/spartan3 FPGA datasheet.