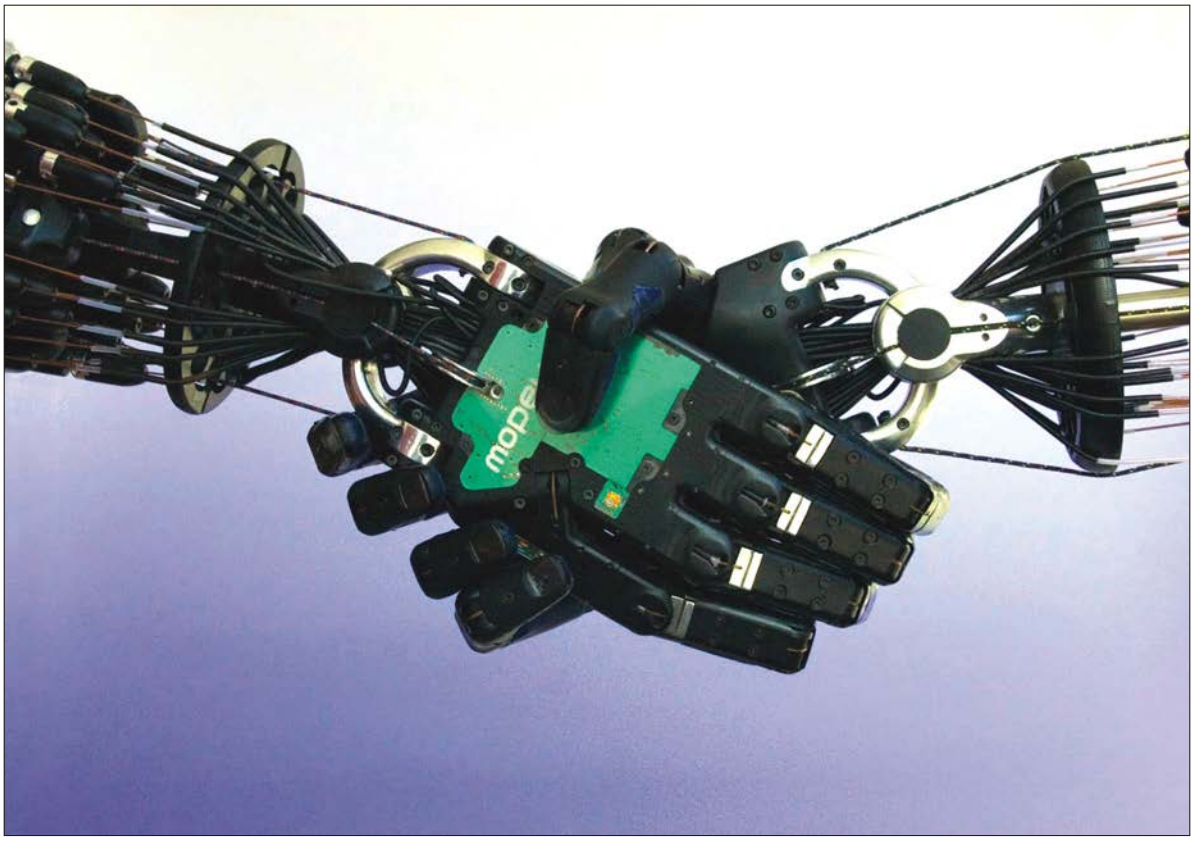

Journal of Artificial Intelligence

ISSN 1994-5450

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Artificial Intelligence 5 (4): 221-226, 2012
ISSN 1994-5450 / DOI: 10.3923/jai.2012.221.226

FPGA Implementation of Audio Enhancement Using Adaptive LMS Filters

[1]V. Elamaran, [2]A. Aswini, [2]V. Niraimathi and [2]D. Kokilavani
[1]School of EEE, SASTRA University, Thanjavur, Tamil Nadu, India
[2]Department of ECE, School of EEE, SASTRA University, Thanjavur, Tamil Nadu, India

Corresponding Author: V. Elamaran, School of EEE, SASTRA University, Thanjavur, Tamil Nadu, India

ABSTRACT

Digital audio has become very popular in the last two decades with the growth of multimedia systems and the World Wide Web. So, audio processing techniques such as filtering, equalization, noise suppression, compression, addition of sound effects and synthesis become necessary in the field of sound engineering. This study has presented some of the audio enhancement techniques using adaptive Least Mean Square (LMS) filters with the Field Programmable Gate Array (FPGA) Architectures which are developed using Xilinx System Generator (XSG). Verilog descriptions from XSG are synthesized to the target FPGA device-a Virtex4 xc4vsx55-12ff1148 and the resource utilization summary for the various alternate LMS architectures is obtained along with the Signal-to-Noise Ratio (SNR) calculations. Results show that Delayed LMS architectures provide a better SNR improvement at the cost of more resource utilizations.

Key words: Adaptive filters, FPGA implementation, LMS algorithm, Xilinx system generator, signal to noise ratio

INTRODUCTION

Audio engineering encompasses the recording, storage, transmission and reproduction of signals to which people listen. In an audio storage and transmission system, the quality of the signal may degrade due to various reasons. For example, low quality speech production system would produce a poor quality audio (Mandal, 2002). The presence of background noise introduced during audio compression is another source of degradation (Manikandan and Madheswaran, 2007). Audio enhancement algorithms can be used to reduce the noise contained in a signal and improve the quality of the audio signal. In this study, digital filtering technique is presented. If the noise component in a signal has a narrow spectrum, a straightforward digital filtering can be applied to suppress the noise components (Ownby and Mahmoud, 2003).

A digital filter (Ramesh and Reddy, 2006) which has self-adjusting characteristics is known as an adaptive filter. It is capable of adjusting automatically its filter coefficients to adapt the input signal via an adaptive algorithm. They play an important role in recent Digital Signal Processing (DSP) products in areas like noise cancellation, biomedical signal enhancement, active noise control, adaptive control systems (Shitong *et al*., 2005), equalization of communication channels and telephone echo cancellation (Tan, 2007). Adaptive filters generally work for spectral overlap between noise and signal, adaption of signal-changing environments and unknown or time-varying noise (Sivakumar *et al*., 2006). For example, when interference noise spectrum which is strong that overlaps the desired signal, the traditional filtering approach will fail to retain the

desired signal spectrum while removing the noise content using a conventional notch filter as shown in Fig. 1. In such a scenario, an adaptive filter will do the filtering successfully but not with a filter that has fixed coefficients.

A simple example using a one-tap adaptive filter is studied as shown in Fig. 2. The desired speech s(n) is captured with the first microphone with analog-to-digital converter (ADC). But the signal is contaminated due to the noisy environment and the ADC produces a signal with the noise; that is d(n) = s(n)+n(n). Only noise is picked up by the second microphone and the second ADC channel captures noise x(n), which is fed to the adaptive filter (Tan, 2007).

A digital filter with adjustable filter coefficient i.e., a one-tap adaptive filter and the LMS algorithm to modify the value of the coefficient while filtering each sample. It produces an estimate of noise, y[n] which is subtracted from the noise contaminated signal d[n] = s[n]+n[n]. When, the estimate of noise equals the n[n] or approximates, i.e., y[n] = n[n], the error signal e[n] = s[n]+n[n]-n[n], which is approximately s[n], the filtered speech signal.

The filter coefficient is adjusted in the LMS algorithm based on the Eq. 1 as follows:

$$w_{n+1} = w_n + \mu\, e[n] \times [n] \tag{1}$$

where, μ is a scalar called step size. Apart from an optimal solution, an another practical advantage of the LMS algorithm (Ilyas *et al.*, 2010) is having only one multiplication and one addition for updating the coefficient. To minimize the mean square energy of the overall output, these weights are adjusted.

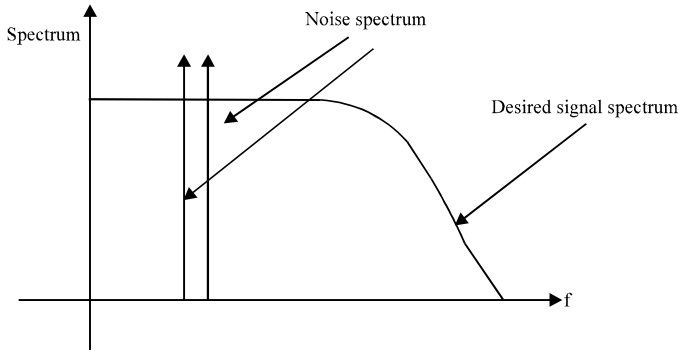

Fig. 1: Spectrum illustration-need for adaptive filters

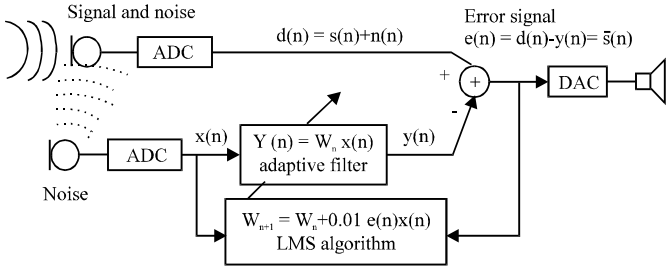

Fig. 2: One-tap adaptive filter

We presented standard LMS architecture using direct form and canonical form to enhance the audio in which the noise is contaminated. Also the delayed LMS architecture is implemented using direct form, canonical form and hybrid form using Xilinx System Generator and the results are obtained in terms of resource utilization summary. Also, SNR calculations for both the input audio and the filtered audio are compared and thus, the SNR improvement is produced.

LMS ARCHITECTURES

The standard LMS algorithm according to the Eq. 1 is implemented as shown in Fig. 3. The model which is implemented using Xilinx System Generator is shown in Fig. 4. The μ value is considered as 0.01 in all the models which are implemented here.

To reduce the critical path, a non canonical LMS architecture in which the transpose FIR is used as shown in Fig. 5.

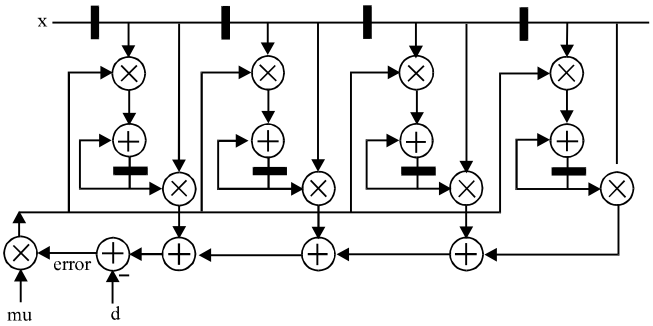

Fig. 3: LMS Implementation-direct form

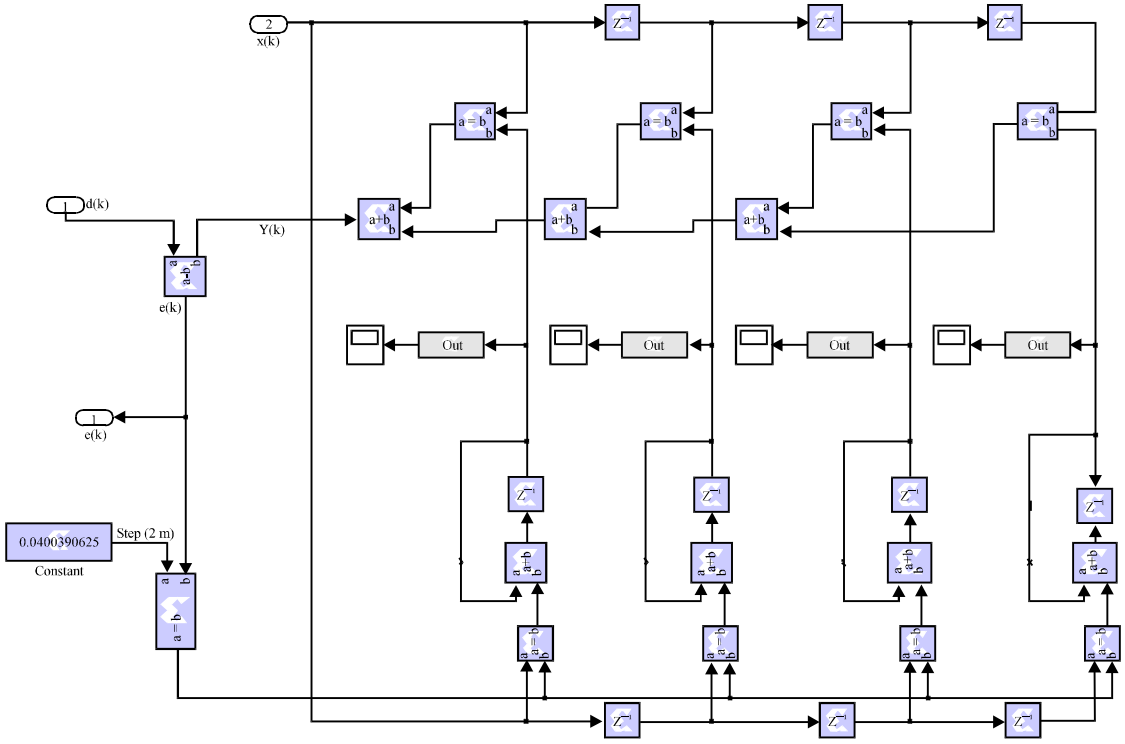

Fig. 4: XSG blocks-direct form LMS architecture

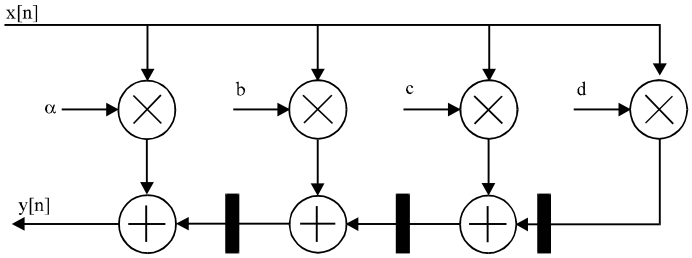

Fig. 5: A non-canonical FIR architecture

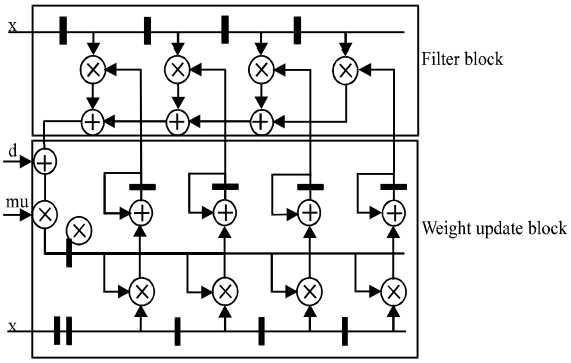

Fig. 6: Delayed LMS-direct form architecture

DELAYED LMS ARCHITECTURES

An implementation of the direct LMS architecture which can be pipelined and retimed to give a functionally equivalent representation named as Delayed LMS (DLMS) (Bahoura and Ezzaidi, 2009) which is shown in Fig. 6.

The transposed DLMS can be obtained by applying retiming to this architecture as was implemented in the case of direct LMS in Fig. 4. This transposed form has a filter block identical to the FIR with the fixed coefficients shown earlier. Figure 7 shows the architecture of transposed DLMS architecture. Figure 7b shows the combination of both yield a hybrid architecture which are faster than direct forms and pipelines can be easily implemented.

Xilinx system generator tool: System generator tool developed by Xilinx which is used to convert the MATLAB/Simulink generated output of any DSP component with testing environment into an Hardware Description Language (HDL) code such as Verilog or VHDL. These codes can be further implemented into any FPGA target devices with the use of Xilinx ISE environment. Here the Verilog descriptions are synthesized in to the target FPGA device (Saadi *et al*., 2007), Virtex4 xc4vsx55-12ff1148 and the SNR results and a resource utilization summary report are obtained in the Table 1 and 2.

System Generator token (Xilinx Corp, 2001) serves as a control panel to control the system and to handle the simulation parameters and also the code generation may be invoked for netlisting. A System Generator token must be added to every Simulink model by which the specifications of code generation and simulation parameters can be modified. The HDL file can be simulated, optimized, synthesized and implemented to the target FPGA device.

Table 1: SNR calculations and SNR improvement

LMS architecture	Output SNR (dB)	SNR improvement (dB)
Direct form LMS	8.880	6.770
Transpose form LMS	9.210	7.100
Delayed LMS	9.520	7.410
Delayed transpose LMS	9.549	7.439
Delayed hybrid LMS	9.548	7.438

Table 2: Resource utilization summary

LMS architecture	Slices	FFs	LUTs	IOBs	DSP48s
Direct form LMS	304	240	304	74	18
Transpose form LMS	388	416	352	98	18
Delayed LMS	456	512	412	98	18
Delayed transpose LMS	472	416	380	98	18
Delayed hybrid LMS	470	448	378	98	18

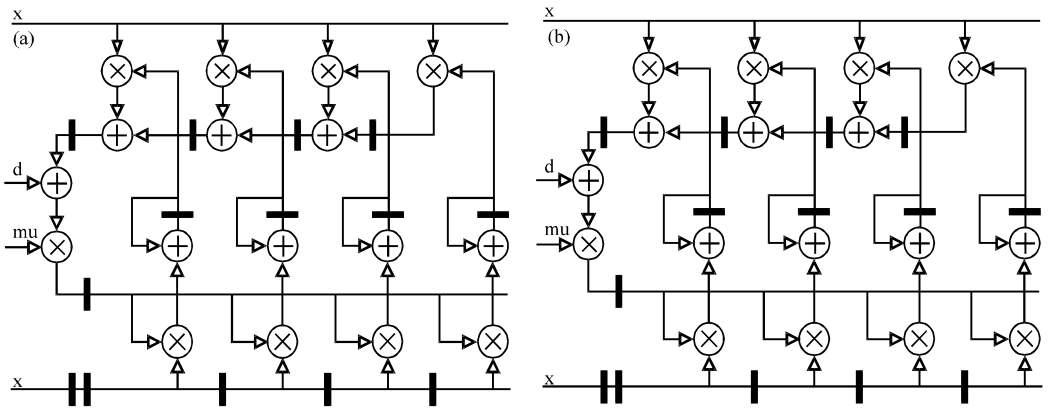

Fig. 7(a-b): (a) Transposed DLMS and (b) hybrid DLMS

SIMULATION RESULTS

An input audio 'drop.wav' , which has sampling rate 48 kHz, 2 channel and 16 bits per sample is used for all the XSG models presented here. A noise is contaminated in to the original audio and the enhanced audio using adaptive LMS architectures is obtained.

Signals-to-noise ratio: The SNR metric is obtained for various models can be calculated as follows in the Eq. 2:

$$\text{SNR} = 10\times\log 10\left(\frac{\text{Signal energy}}{\text{Noise energy}}\right) \tag{2}$$

where, the noise energy is common for the calculations of input SNR and output SNR. Only the signal energy is enhanced in the output section and the output SNR is calculated and the results are shown in Table 1. The input SNR is evaluated as 2.11 dB and it is subtracted from the output SNR to determine the SNR improvement for all the models.

Resource utilization: Device resources utilization summary for the implementation model can be quickly estimated by the block named Xilinx Resource Estimator and the report is obtained in the Table 2.

CONCLUSION

High performance DSP algorithms are handled by modern FPGAs, but the lack of knowledge behind FPGA configuration with DSP concepts slowed their wider adoption. Using System Generator, digital designers may efficiently adopt DSP applications into digital designs using simple quicker and easy Simulink XSG design methodology. We have demonstrated audio enhancement techniques by various alternative LMS implementations which are investigated and the results show that the delayed LMS architectures obtain a better SNR improvement at the cost of more resource utilizations.

ACKNOWLEDGMENTS

It is a great pleasure for me to express my deep gratitude and sincere thanks to Dr. Har Narayan Upadhyay, Associate Dean and Dr. R. John Bosco Balaguru, Associate Dean, Electronics and Communication Engineering Department, School of EEE, SASTRA University, Thanjavur for their constant encouragement and valuable guidance during our research work.

REFERENCES

Bahoura, M. and H. Ezzaidi, 2009. FPGA Implementation of a Sequential Adaptive Noise Canceller using Xilinx System Generator. Proceedings of the Conference on Microelectronics, December 19-22, 2009, Marrakech, pp: 213-216.

Ilyas, M.Z., S.A. Samad, A. Hussain and K.A. Ishak, 2010. Improving speaker verification in noisy environments using adaptive filtering and hybrid classification technique. Inform. Technol. J., 9: 107-115.

Mandal, M.K., 2002. Multimedia Signals and Systems. Kluwer Academic Publishers, Netherlands.

Manikandan, S. and M. Madheswaran, 2007. Implementation of active noise filter for real-time noise reduction using the TMS320C5402 DSP kit. J. Applied Sci., 7: 2750-2754.

Ownby, M. and W.H. Mahmoud, 2003. A design methodology for implementing DSP with Xilinx system generator for MATLAB. Proceedings of the 35th Southeastern Symposium on System Theory, 16-18 March 2003, Cookeville, TN, USA., pp: 404-408.

Ramesh, R. and P.S. Reddy, 2006. An elegant and simple method to test the stability of 2-D recursive digital filters. Inform. Technol. J., 5: 1068-1072.

Saadi, S., M. Touiza and A. Guessoum, 2007. FPGA implementation of 2D signals encoder using QMF based dyadic DWT: Application to neutron tomography projections. J. Applied Sci., 7: 1528-1533.

Shitong, W., L. Yueyang, F.L. Chung and C. Shu, 2005. Iterative self-adaptive filtering algorithm for reducing impulsive noise in color images. Inform. Technol. J., 4: 456-461.

Sivakumar, B., S. Subha Rani, V.S.V. Ravi Kiran and P. Abhishek, 2006. Adaptive filter approaches for interference suppression in CDMA systems. Inform. Technol. J., 5: 1098-1101.

Tan, L., 2007. Digital Signal Processing: Fundamentals and Applications. Academic Press, New York, USA., Pages: 840.

Xilinx Corp, 2001. Xilinx system generator version 2.1 for Simulink. Xilinx Development System, USA.