Journal of Electron Devices, Vol. 17, 2013, pp. 1423-1426

© JED [ISSN: 1682 -3427]

Journal of Electron Devices
www.jeldev.org

FPGA BASED HIGH FREQUENCY NOISE ELIMINATION SYSTEM FROM SPEECH SIGNAL USING XILINX SYSTEM GENERATOR

Prasit Kumar Bandyopadhyay[1], Arindam Biswas[1], Pramit Kumar Bandyopadhyay[2] , Durbadal Mandal[3], Rajib Kar[3]

[1]Dumkal Institute of Engineering & Technology, W. B. University of Technology, Dumkal-742406, Dist. Murshidabad, India
[2] Wipro Technologies, 72 Keonics, Electronics City, Wipro Avenue, Bangalore-100, India.
[3]Dept. of Electronics and Communication Engineering, N. I. T., Durgapur-713209, India.
mailarindambiswas@yahoo.co.in

Received 14-01-2013, online 28-01-2013

ABSTRACT
This paper introduces a new method for eliminating high frequency noise from audio speech signals. Current noise reduction techniques have generally proven to be effective, yet these typically exhibit certain undesirable characteristics. Distortion and/or alteration of the audio characteristics of primary audio sound is a common problem. Also user intervention in identifying the noise profile is sometimes necessary. The proposed technique is centered on the MAC FIR filtering technique for noise removal but uses a novel architecture whereby advanced signal processing techniques are used to identify and preserve the richness of the audio spectrum by using Xilinx System Generator.
Keywords: MAC FIR filtering, Xilinx System Generator, Dual Port Rom.

I. INTRODUCTION

Recent work on FPGA based apllication is already exits [1-5] in literature of VLSI technology. Speech is natural mode for communication and is highly efficient. Of all manmade sounds which influence our lives, speech and music are arguably the most prolific. Speech has long been perceived as natural interface between peoples and computers and hence has received much focused attention. In this paper noise extraction from speech signal is given[2-8].

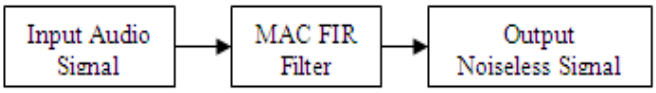
Rapid advances made in VLSI and WSI technologies have led the way to an entirely different approach to computer design for real-time applications, using special-purpose architectures with custom chips. By migrating some of the highly compute-intensive tasks to special-purpose hardware, performance which is typically in the realm of supercomputers, can be obtained at only a fraction of the cost. High-level Computer-Aided Design (CAD) tools for VLSI silicon compilation, allow fast prototyping of custom processors by reducing the tedium of VLSI design.

The IIR filters are not well supported by softwares and Intellectual Property cores as compared to the FIR filters [7].

In this paper, we discuss the implementation of FIR Bandpass Filters on FPGAs which will eliminate high frequency noise from audio speech signal.

II. BLOCK REPRESENTATION:

The block representation of the noise extraction system is given below



III. SYSTEM GENERATOR

Xilinx produces many software tools to simplify the design task for hardware programmers. One such program is DSP System Generator. It interfaces with Matlab Simulink and provide efficient implementations of digitally realizable Simulimk blocks and commonly used digital programming blocks. Another important feature of System Generator is the Gateway In/ Out blocks. These provide the interface between the double precision of Simulink with the floating point architecture of FPGA.

IV. INPUT AUDIO SIGNAL

An 8000 Hz and a 22050 Hz 16 bit mono audio signal in .avi format are taken as inputs. The audio files are taken in .avi (Audio Video Interleaved) format for better quality. In the simulation window of Xilinx System Generator software, a 'From multimedia Block' is taken from simulink library browser. The 'from multimedia file' block reads audio samples from a multimedia file and imports them into a simulink model. The audio file is un buffered by a 'un buffer' block and is then fed to the input of the system.

V. FIR FILTER

Finite Impulse Response **(**FIR**)** filter is a type of a signal processing filter whose impulse response (or response to any finite length input) is of *finite* duration, because it settles to zero in finite time. This is in contrast to infinite impulse response (IIR) filters, which have internal feedback and may continue to respond indefinitely (usually decaying). The impulse response of an Nth-order discrete-time FIR filter lasts for N+1 samples, and then dies to zero
In this design a MAC (Multiplier Accumulator) based FIR filter has been used.

VI. MAC (MULTIPLIER ACCUMULATOR)

A 12-bit x 8-bit MAC (Multiplier Accumulator) with a multiplier and an accumulator has been designed using Multiplier and Accumulator block taken from Simulink Library Browser.
VI. 1 Multiplier: The 'Mult' Block taken from simulink library browser implements multiplier. It computes the product of the data on its two input port and produces the result on its output port. The block supports a size-performance tradeoff in its implementation. It can be used either as a parallel multiplier that operates on the full width data (faster and larger) or as a sequential multiplier that computes the result from smaller partial products (slower and smaller). It has been used here as a parallel multiplier and the latency of the multiplier has been set to 3. Multiplier input data widths of 12 bits and 8 bits of signed data and output data width of 20 bits is taken.
VI.2 Accumulator: The accumulator block implements an adder or substractor based scaling factor. The block's current input is accumulated with a scaled current stored value. The block has an input b and an output q. The output will have same arithmetic type and binary point position as the input. The q is calculated as follows

$$q(n) = \begin{cases} 0 & if\ rst=1 \\ q(n-1)\,x\,Feedback\&aling \pm b(n-1) & otherwise \end{cases}$$

Here the accumulator has a output width of 27 bits. The block always has a latency of 1 sample period.

VII. DUAL PORT RAM

In a typical MAC FIR the coefficients and data must be stored in a memory system. There are

several storage options such as block RAM, distributed RAM and SRL16E. Dual port block RAM is used here to store data and coefficients, with the data being captured and read out in a cyclic data RAM buffer. Therefore the RAM used here is in mixed mode configuration. The data is written and read from port A (RAM mode) and the coefficients are read only from port B (ROM mode).

VIII. PADDING & UNPADDING:

Each port width of the dual port block RAM is determined by its respective input width and the ports can only be different if the widths are 2, 4, 8, 16 or 32 times larger in respect to each other. As the desired width do not agree with this rule, they will have to be the same. There the data must be manipulated before and after the RAM. On the input, padding the 8-bit input data to 12 bits will match up the port widths. The output can be readjusted to the original 8- bits by using unpadding. This concept is illustrated in figure below

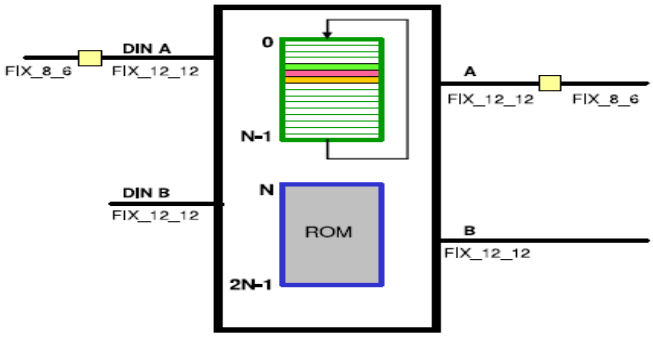


Fig.1: Padding & Unpadding

IX. ADDRESS GENERATOR

An address generator for 92- tap MAC based FIR filter is designed by using predefined functionality blocks. This process includes creating the signal that drives, the address for the memory in which the sample will reside (data_addr) and the address for the memory in which the coefficient will reside (coef_addr); and the write enable signal which dictates when a new sample can be saved in the memory (we).

X. RESULTS & DISCUSSION

This paper describes about the Noise Extraction system that is designed in Xilinx System Generator software is used to eliminate high frequency noise from audio signal. Here the outputs of the system are studied. The results which are obtained are categorized as synthesize report or software simulation result and hardware co simulation result.

X.1 Software Simulation: The output which are obtained after simulating the design in system generator software environment is given below

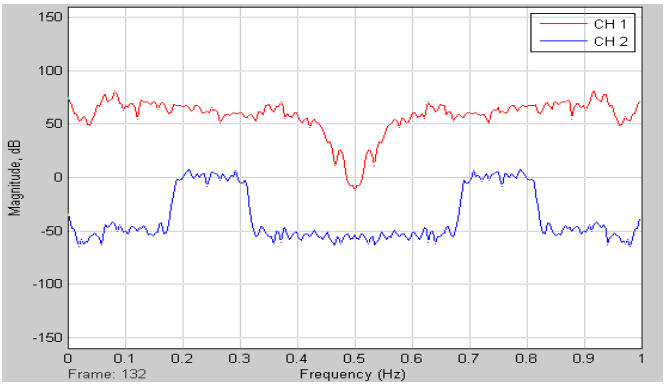


Fig 2: Output for 22050 Hz Audio Signal
CH1: Signal with noise
CH2: Signal without Noise

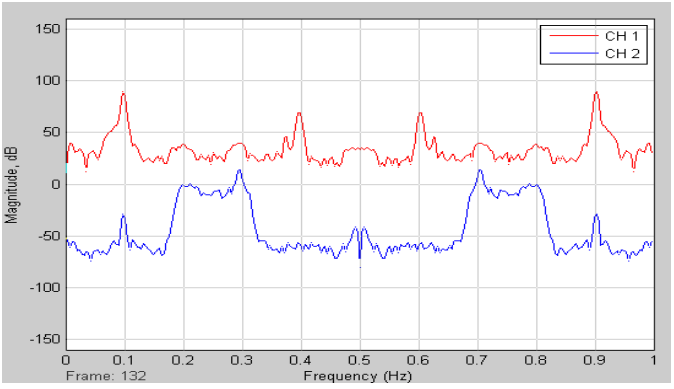


Fig 3: Output for 8000 Hz Audio Signal
CH1: Signal with noise
CH2: Signal without Noise

X.2 Hardware Co Simulation: Hardware Co-Simulation is a great feature in System Generator that allows users to run the full or part of the

System Generator design to FPGA and increase the simulation speed dramatically. System Generator already includes HWCOSIM plugins for commonly used DSP demo boards. It is possible to use the System Generator Board Description Builder (SBD Builder) to create new HWCOSIM plugins for unsupported boards. Hardware Co- Simulation of the entire System Generator design is done using Spartan 3E FPGA kit. The outputs which are obtained during hardware simulation and software simulation are same. Which indicates the hardware simulation is successful.

Currently, the FPGA technology has been advanced enough to model complex chips with the realistic operating frequency. A simulator runs on an off- the-shelf machine with one or several parts of the simulator implemented on FPGA. Whenever the simulation reaches a point where the simulator's functions are implemented in the FPGA, the simulator interacts with the FPGA.

XI. CONCLUSION

The report discussed the design and test results of FPGA (Field Programmable Gate Array) based Noise Extraction System for extraction of High frequency noise from audio speech signal. The goal is to develop a software package that processes audio speech signal and extracts noise from that signal and to implement the design on FPGA. The project successfully achieved the implementation of the design to FPGA and the HIL (Hardware-in-the-loop) verification for the design; however the real time verification of the output of the FPGA was not accomplished.

The entire design is done in the Xilinx System Generator Software environment using Xilinx DSP (Digital Signal Processing) Block-set. Digital Signal Processing allows much greater sophistication to be applied to noise reduction. In addition DSP allows workstation style editing operations to take place to actually splice and rebuild damaged portion of a program segment.

Two different audio signals are taken as input from which the noise has been extracted. The signals are recorded and taken in avi (audio video interleaved) format rather than wav format for better quality. The entire system consists of two phases, a filtering phase and an implementation phase. In filtering phase the input signal is filtered using a MAC (Multiplier Accumulator) based FIR Bandpass filter. The coefficients for the Bandpass filter are generated by specifying the passband. In the implementation phase the design is implemented on the FPGA Spartan 3E kit by generating the VHDL code for the design.

Acknowledgement

The authors would also like to thank Depanwita Debnath, Asst. Professor, Techno India University Salt Lake for many useful discussion during the computational work.

References

[1] Amrita, Rajesh Mehra, "Embedded Design of an Efficient Noise Canceller for Digital Receivers", International Journal of Engineering Science & Technology (IJEST), **3**, 1252 (2011).

[2] Mahdi Shaneh, and Azizollah Taheri, "Voice Command Recognition System Based on MFCC and VQ Algorithms", World Academy of Science, Engineering and Technology , **33**, 534 (2009).

[3] Fábio Fabian Daitx, Vagner S. Rosa, Eduardo Costa, Paulo Flores, Sergio Bampi, " VHDL Generation for Optimized FIR Filters", 2008 International Conference on Signals, Circuits and Systems organized by Informatics Institute, UFRGS, Porto Alegre, RS, Brazil.

[4] Juan J. Rodriguez-Andina, Maria J. Moure, and Maria D. Valdes, Features, "Design Tools, and Application Domains of FPGAs", IEEE Transactions on Industrial Electronics, **54**, 1810 (2007)

[5] K. Sridharan and T. K. Priya, "The Design of a Hardware Accelerator for Real-Time Complete Visibility Graph Construction and Efficient FPGA Implementation", IEEE Transactions on Industrial Electronics, **52**, 1185 (2005).

[6] Mohd Fadzil Ain, Majid S. Naghmash, Y. H. Chye, "Synthesis of HDL Code for FPGA Design Using System Generator", European Journal of Scientific Research, **45**, 111 (2010).

[7] Louis Litwin, "FIR and IIR Digital Filters", IEEE potentials, **19**, 28 (2002)

[8] L. Rabiner, and B. H. Juang, "Fundamentals of Speech Recognition", Prentice-Hall, New Jersey (1993).