Available online at www.sciencedirect.com

ScienceDirect

Procedia Technology 11 (2013) 1003 – 1010

Procedia Technology

The 4th International Conference on Electrical Engineering and Informatics (ICEEI 2013)

Design and Implementation of Real Time Noise Cancellation System Based on Spectral Subtraction Method

Anugerah Firdauzi*, Kiki Wirianto, Muhammad Arijal, Trio Adiono

School of Electrical Engineering and Informatics, Bandung Institute of Technology, Indonesia

Abstract

In this paper, a real-time digital signal noise cancellation system is designed based on Spectral Subtraction Method. The system cancels the noise in frequency domain by estimating the noise energy spectral from a noisy input. Since that method is performed in frequency domain with polar form, a time to frequency domain and rectangular to polar transformer with each of their complement circuits are needed to perform the noise cancellation operation. Considering high speed computation with lower system frequency for a real-time processing and low power consumption, an FPGA based full-hardware implementation is used in proposed system. As a result, the proposed noise cancellation system implementation result is able to perform a real time noise cancellation with the SNR value of 71 dB and use about 47,000 FPGA logic elements with 32 bits data resolution.

Selection and peer-review under responsibility of the Faculty of Information Science & Technology, Universiti Kebangsaan Malaysia.

Keywords: Noise cancellation; spectral subtraction; real-time system, full-hardware implementation

1. Introduction

The proposed noise cancellation algorithm is designed based on Spectral Subtraction Method from [1]. This method cancels the noise based on energy spectral of the noisy speech signal. The block diagram of the algorithm is shown in Fig. 1.

* Corresponding author.
E-mail address: tadiono@gmail.com

Selection and peer-review under responsibility of the Faculty of Information Science & Technology, Universiti Kebangsaan Malaysia.
doi:10.1016/j.protcy.2013.12.287

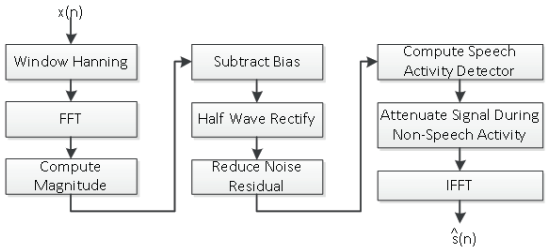

Fig. 1. Noise cancellation algorithm

From [1], signal sampled *k* and noisy speech signal $x(k)$ can be modeled as the sum of speech signal $s(k)$ and noise signal $n(k)$. The FFT is used for analyzing the spectral noisy speech signal. This leads to equations

$$x(k) = s(k) + n(k) \tag{1}$$

$$X\left(e^{j\omega}\right) = FFT\left(x(k)\right) = \sum_{k=0}^{L-1} x(k)e^{-j\omega k} \tag{2}$$

where *L* is the number of FFT points. The relation between input and output of noise cancellation system are

$$\hat{S}\left(e^{j\omega}\right) = FFT\left(\hat{s}(k)\right) = H\left(e^{j\omega}\right)X\left(e^{j\omega}\right) \tag{3}$$

where $H(e^{j\omega})$ is the spectral subtraction filter and $\hat{S}(e^{j\omega})$ is the estimated spectral of speech signal.

The noise $\widehat{N}\left(e^{j\omega}\right)$ can be estimated and measured during non-speech activity with the length of noise frame *M* by averaging its spectral magnitude. The estimated speech signal can be modeled as a subtraction of magnitude of noisy speech signal and magnitude of estimated noise. It gives the equation

$$\mu\left(e^{j\omega}\right) = E\left\{\left|\widehat{N}\left(e^{j\omega}\right)\right|\right\} = \frac{1}{M}\sum_{i=0}^{M-1}\left|X_i\left(e^{j\omega}\right)\right| \tag{4}$$

$$\hat{S}\left(e^{j\omega}\right) = \left[\left|X\left(e^{j\omega}\right)\right| - \mu\left(e^{j\omega}\right)\right]e^{j\theta_x\left(e^{j\omega}\right)} \tag{5}$$

where $X_i\left(e^{j\omega}\right)$ is the *i*-th noisy signal frame to be considered as noise during non-speech activity and $\theta_x\left(e^{j\omega}\right)$ is the phase of $X\left(e^{j\omega}\right)$. From equation (3) and (5), the spectral subtraction filter will be

$$H\left(e^{j\omega}\right) = 1 - \frac{\mu\left(e^{j\omega}\right)}{|X(e^{j\omega})|} \tag{6}$$

Half wave rectification is needed to decrease noise tone, but in the other hand, can remove the speech information incorrectly. To implement half wave rectification, spectral subtraction filter can be modified into

$$H_R\left(e^{j\omega}\right) = \frac{H\left(e^{j\omega}\right) + \left|H\left(e^{j\omega}\right)\right|}{2} \tag{7}$$

and then the term $H_R(e^{j\omega})$ is the new spectral subtraction filter.

From [1], the speech activity can be detected by comparing the next, present and the previous value of spectral magnitude of the estimated speech. If the spectral magnitude of the estimated speech is higher than the maximum spectral magnitude of the estimated noise, there is high chance that the frame is speech. If the spectral magnitude of the estimated speech is lower than the maximum spectral magnitude of the estimated noise, but has a nearly constant value, there is high chance that the frame is due to low energy speech. If the spectral magnitude of the estimated speech is lower than the maximum spectral magnitude of the estimated noise and varies by frame to frame, there is high chance that the frame is due to noise. For *i*-th estimated signal frame, this algorithm can be implemented as

$$\left|\hat{S}_i(e^{j\omega})\right| = \left|\hat{S}_i(e^{j\omega})\right| \text{if } \left|\hat{S}_i(e^{j\omega})\right| \geq max\left|\widehat{N}_R(e^{j\omega})\right| \tag{8}$$

and

$$\left|\hat{S}_i(e^{j\omega})\right| = min\left\{\hat{S}_{i+1}(e^{j\omega}), \hat{S}_i(e^{j\omega}), \hat{S}_{i-1}(e^{j\omega})\right\}, \text{if } \left|\hat{S}_i(e^{j\omega})\right| < max\left|\widehat{N}_R(e^{j\omega})\right| \tag{9}$$

Implementing this part of the algorithm can cause a frame delayed output due to the next frame data.

From [1], the frame is considered as a speech if the power ratio is above at least 12 dB. This leads to

$$\hat{S}_i(e^{j\omega}) = \begin{cases} \hat{S}_i(e^{j\omega}), & T \geq -12dB \\ 0, & T < -12dB \end{cases} \tag{10}$$

2. The System Implementation

The proposed noise cancellation system is implemented into full logic circuit to obtain a very high speed processing time. The proposed system architecture is shown in Fig. 2(a) while the detail block diagram is shown in Figure 2(b). The system uses two different clock frequencies. The first clock is 12 kHz clock which is used for interfacing between FPGA and the audio codec. The other clock is 10 MHz clock which is used for data.

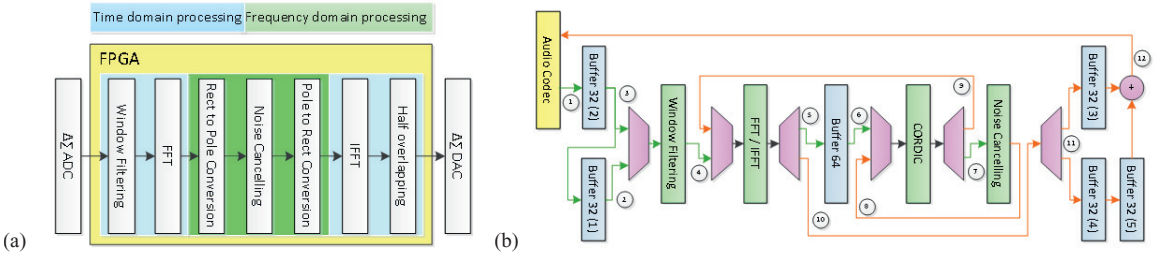

Fig. 2. Noise cancellation system (a) architecture and (b) circuit structure

The timing diagram of data processing can be seen in Fig. 3. The figure shows the activity of each sub circuit in the system, its working mode, and the number of required clock cycles for each operation. The system requires 384 clock cycles to complete the noise cancellation function.

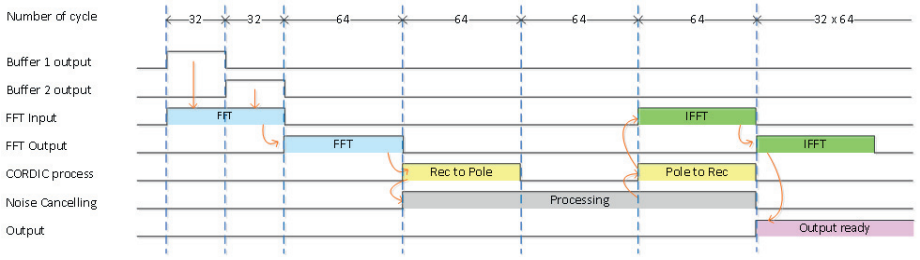

Fig. 3. Noise cancellation system timing diagram

2.1. FFT and IFFT Circuit

The FFT circuit used in this system is 64 points FFT with serial input. This FFT circuit has a synchronous stream of input data in every clock cycle for a total of 64 cycles. The synchronized input is handled by the input buffers, thus the sending of the FFT data to the system is done in every time of the availability of the input buffer.

The FFT circuit is also used to process IFFT by simply inverting the imaginary part of the twiddle factor and dividing the output by the number of FFT points. The block diagram of FFT circuit is shown in Fig. 4.

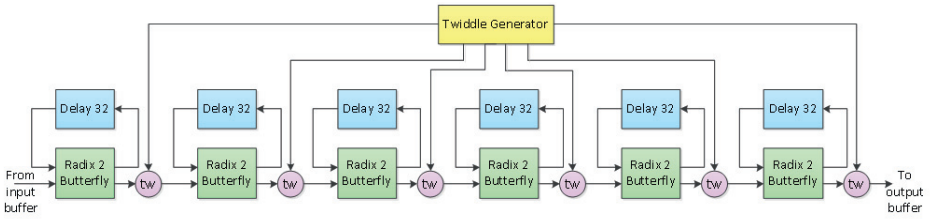

Fig. 4. The 64 points serial FFT circuit block diagram

2.2. CORDIC Circuit

CORDIC (COordinate Rotation DIgital Computer) is hardware-efficient trigonometric algorithms that provides iterative solutions for trigonometric and other transcendental functions by using only shifts and add operations. For the proposed system, CORDIC is used to perform rectangular to polar conversion and vice versa. The architecture of CORDIC is 32-bit unrolled pipelined CORDIC processor which has 29 iterations and 5-stage pipeline (using 4 registers). The block diagram of the CORDIC circuit is shown in Fig. 5.

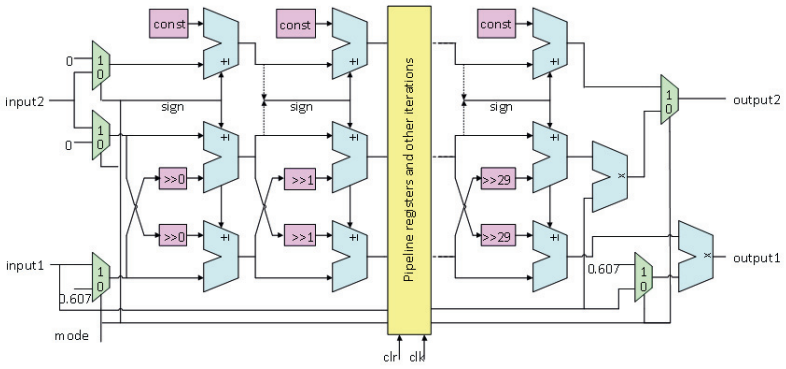

Fig. 5. Pipelined Unrolled CORDIC circuit block diagram

2.3. The Noise Cancellation Circuit

The noise cancellation circuit works in 2 modes: noise sampling and noise cancellation. The noise sampling mode calculates the value of the noise and stores the average value and maximum value to mean and max memory respectively. The noise cancellation stage is performed in 2×64 cycles. The first stage is the calculation of the next value; the second stage is the calculation of the present, previous, and temporary memory value; while the last stage is the calculation of the output data. The data flow of this circuit is based on the algorithm presented in Figure 1. The block diagram of the noise cancellation system is presented in Fig. 6.

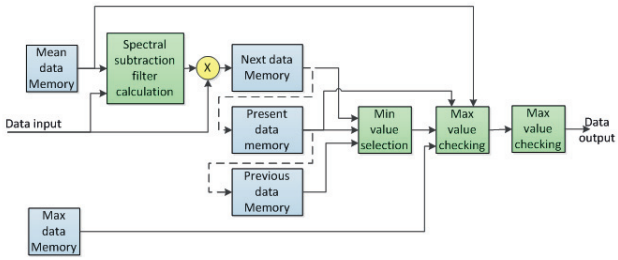

Fig. 6. The noise cancellation circuit block diagram

3. System Verification

In order to verify system functionality and measure the performance results, we perform both system simulation and a real-time verification. Simulation is done using MATLAB® software.

3.1. System Simulation

The noisy speech signal, estimated noise, and the estimated speech from the algorithm described above are plotted in time domain to be compared with other algorithm. The plotted data in time domain are presented as follows (Fig. 7(a) to 7(e)):

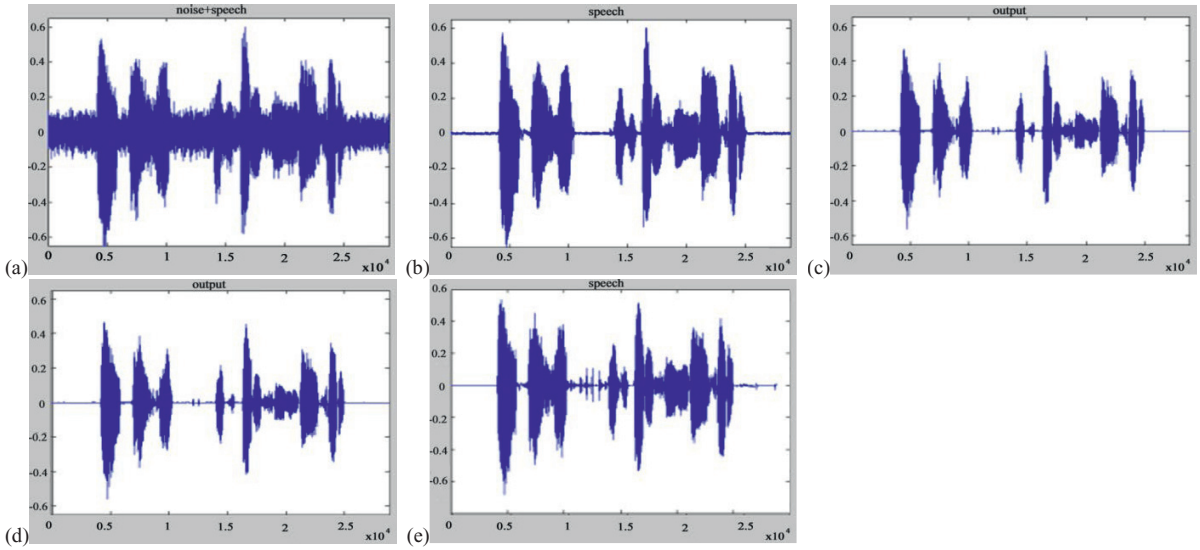

Fig. 7. (a). Noisy speech input signal, (b). Clean speech signal, (c). Output signal with Spectral Subtraction Method based on Boll Algorithm, (d). Output signal with Spectral Subtraction Method based on Boll Algorithm + Monte Carlo Noise Localization, (e). Output signal with Modified Spectral Subtraction Method with a weighting function and scaling factor

Signal to noise ratio (SNR) of input system (SNR_i) is calculated based on ratio of the power of the pure signal sample to the noise signal sample, whether SNR of output system (SNR_o) is calculated based on the ratio of power of the output signal to the power of detected noise signal by Voice Activity Detector. The simulation results in MATLAB, for SNR_i = 6.4151 dB, are presented in the Table 1. From the simulation results, we can conclude that the spectral subtraction algorithm based from Boll algorithm is extremely good at noise suppression and estimating higher energy speech, but may have a problem with the lower energy speech. The approach based on weighting function and scaling factor is not good enough because there are some residual noise that are not cancelled by the system.

Table 1. SNR_o value comparison for several algorithms

Algorithm	SNRo
Modified SS with weighting function and scaling factor	56.53 dB
SS – Boll Algorithm + Monte Carlo Noise Localization	71.47 dB
SS – Boll Algorithm	**71.29 dB**

The Combination of SS Method based on Boll algorithm and Monte Carlo Noise Localization approach is a better approach for cancelling a non-deterministic noise because it can learn the noise over the time, but not good to be implemented in full-hardware because it needs extra resources (logic elements). Based on this analysis, it can be concluded that the approach based on Boll algorithm is a better approach that can lead to efficient design with an acceptable level of noise cancellation based on SNR_o.

3.2. Real-time Verification

To observe the noise cancellation result without additional software, a VGA display is implemented in second FPGA to show the signal activity graphically. In this implementation, the first board is implementing the noise cancellation while the second board is used to display the left and right channels containing the result to the monitor. The noisy input signal is sent to the first FPGA for noise cancellation processing. The noise cancellation result is passed to the second FPGA to be displayed and heard through the speaker. The block diagram and realization can be seen in Figure 8(a) and 8(b) respectively.

The demonstration can be seen in the following link: http://www.youtube.com/watch?v=DgQMxt6JRWU

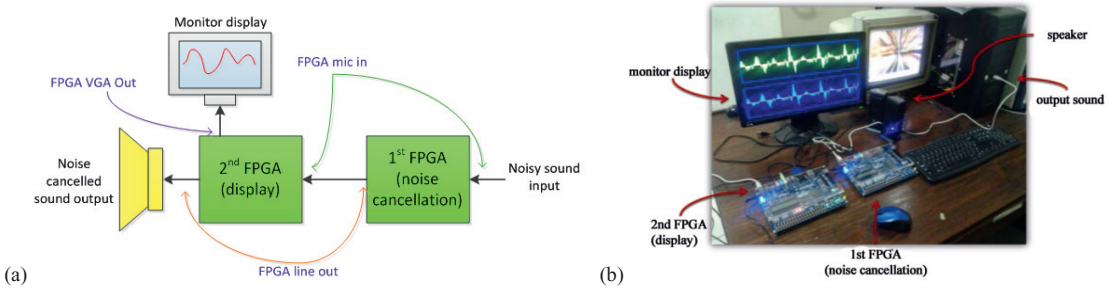

Fig. 8. (a) System application block diagram (b) System application realization

4. The System Performance

The performance of implemented system is measured using its size, speed and voice quality.

4.1. Size and Speed Comparison

50 XOR circuit is a standard comparison circuit to test the design implementation, while hardware-software combination method is another approach in implementing the design by putting the complex noise cancellation process in software (FPGA CPU) for a smooth and easy implementation with high level language.

Table 2 below gives a description about the full-hardware system design compared to basic 50 XORs circuit and hardware-software combination system.

Table 2. Speed and size comparison

System	Logic Elements	Clock Frequency
50 XORs	17	66.54 MHz
Hardware-Software Combination	33100	31.44 MHz
Full hardware	**47000**	**10.00 MHz**

4.2. Qualitative Design Result

The following Fig. 9(a) to 9(c) show the comparison between noisy input signal to the output of our real-time and hardware-software combination noise cancellation system.

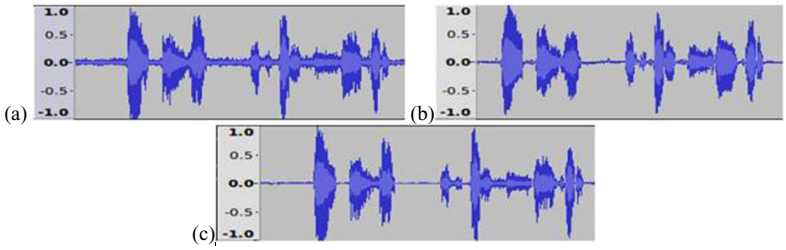

Fig. 9. (a) Noisy input signal waveform (b) Hardware-Software combination noise cancellation system output waveform (c) Full-hardware noise cancellation system output waveform

As can be seen on those pictures, the real-time system has successfully canceled almost all the present noise. Its cancellation result is also good enough compared to the software-hardware combination.

As a result of proposed design and implementation, the system can reach following performance:

a) Real Time Data Processing

Since this noise cancellation system is implemented in full-hardware mode, the timing behavior of the system can be managed to process the data precisely. Thus, it is possible to process data in real time, since the output's delay latency is so small that make it allowable to be neglected.

b) Low Power

The system implemented needs to clocked in a low frequency, about 7 to 10 MHz, to be able to operate well. Since the logic transition will be less if the clock slower, the power consumed will also be less then higher frequency system.

c) High Data Resolution

Each data processing module in this noise cancellation system is implemented for 32 bits data. This wide data range makes the system able to process data with a high resolution. Moreover, this wide data bits will prevent overflow during processing.

5. Conclusion

The real-time noise cancellation hardware system has been successfully implemented using Spectral Subtraction Method and Voice Activity Detector. The implementation of noise cancellation system uses full-hardware approach by directly mapping the complex algorithm into optimized full hardware system architecture considering the design size, high data resolution and low power consumption. The implementation of the design can have SNR value up to 71.29 dB, with 47000 FPGA logic elements, and maximum clock frequency up to 10 MHz.

References

[1] Boll, Steven F.,. *Suppression of Acoustic Noise in Speech Using Spectral Subtraction*. IEEE Transactions on Acoustics, Speech, and Signal Processing. 1979.

[2] Proakis, John G., Manolakis, Dimitris G., *Digital Signal Processing: Principles, Algorithms, and Applications, 4th Edition*. New Jersey: Pearson Prentice Hall. 2007.

[3] Andraka, R., *A survey of CORDIC algorithms for FPGA based computers*. 1998.

[4] Simak, B., Verteletskaya, E., *Noise Reduction Based on Modified Spectral Subtraction Method*. IAENG International Journal of Computer Science, 2011;38:1.