

Article

Pipelined Architecture of Multi-Band Spectral Subtraction Algorithm for Speech Enhancement

Mohammed Bahoura

Department of Engineering, Université du Québec à Rimouski, 300, allée des Ursulines, Rimouski, QC G5L 3A1, Canada; Mohammed_Bahoura@uqar.ca; Tel.: +1-418-723-1986

Received: 29 August 2017; Accepted: 27 September 2017; Published: 29 September 2017

Abstract: In this paper, a new pipelined architecture of the multi-band spectral subtraction algorithm has been proposed for real-time speech enhancement. The proposed hardware has been implemented on field programmable gate array (FPGA) device using Xilinx system generator (XSG), high-level programming tool, and Nexys-4 development board. The multi-band algorithm has been developed to reduce the additive colored noise that does not uniformly affect the entire frequency band of useful signal. All the algorithm steps have been successfully implemented on hardware. Pipelining has been employed on this hardware architecture to increase the data throughput. Speech enhancement performances obtained by the hardware architecture are compared to those obtained by MATLAB simulation using simulated and actual noises. The resource utilization, the maximum operating frequency, and power consumption are reported for a low-cost Artix-7 FPGA device.

Keywords: FPGA; hardware/software co-simulation; pipelining; speech enhancement; multi-band spectral subtraction; signal-to-noise ratio

1. Introduction

The enhancement of speech corrupted by background noise represents a great challenge for real-word speech processing systems, such as speech recognition, speaker identification, voice coders, hand-free systems, and hearing aids. The main purpose of speech enhancement is to improve the perceptual quality and intelligibility of speech by using various noise reduction algorithms.

Spectral subtraction method is a popular single-channel noise reduction algorithm that has been initially proposed for speech enhancement [1]. This basic spectral subtraction method can substantially reduce the noise level, but it is accompanied by an annoying noise in the enhanced speech signal, named musical noise. A generalized version of this method has been proposed to reduce the residual musical noise by an over-subtraction of the noise power [2]. The multi-band algorithm has been developed to reduce additive colored noise that does not uniformly affect the entire frequency band of useful speech signal [3]. Improved version based on multi-band Bark scale frequency spacing has been also proposed to reduce the colored noise [4]. An adaptive noise estimate for each band has been proposed [5]. Furthermore, the spectral subtraction approach has also been applied to other kinds of sounds such as underwater acoustic sounds [6], machine monitoring [7,8], hearing aid [9], pulmonary sounds [10–12], etc.

In real-world applications, such as hands-free communication kits, cellular phones and hearing aid devices, these speech enhancement techniques need to be executed in real-time. Hardware implementation of this kind of algorithms is a difficult task that consists in finding a balance between complexity, efficiency and throughput of these algorithms. Architectures based on the spectral subtraction approach have been implemented on Field Programmable Gate Array (FPGA) devices [13–19]. However, these architectures perform a uniform spectral subtraction over the entire frequency band and, therefore, they do not efficiently suppress colored noise.

In this paper, a new pipelined architecture of multi-band spectral subtraction method has been proposed for real-time speech enhancement. The proposed architecture has been implemented on FPGA using the Xilinx System Generator (Xilinx Inc, San Diego, CA, USA) programming tool and the Nexys-4 (Digilent Inc, Pullman, WA, USA) development board build around an Artix-7 XC7A100T FPGA chip (Xilinx Inc, San Diego, CA, USA). Mathematical equations describing this speech enhancement algorithm (Fourier transform, signal power spectrum, noise power estimate, multi-band separation, signal-to-noise ratio, over-subtraction factor, spectral subtraction, multi-band merging, inverse Fourier transform, etc.) have been efficiently modeled using the XSG blockset. High-speed performance was obtained by inserting and redistributing the pipelining delays.

The rest of the paper is organized as following: Section 2 presents the theory details of the spectral subtraction theory for speech enhancement. Section 3 presents the XSG-based hardware system and discusses the details of different subsystems. Speech enhancement performances are presented in Section 4. Finally, conclusion and perspective are provided in Section 5.

2. Spectral Subtraction Methods

In the additive noise model, it is assumed that the discrete-time noisy signal $y[n]$ is composed of the clean signal $s[n]$ and the uncorrelated additive noise $d[n]$.

$$y[n] = s[n] + d[n] \quad (1)$$

where n is the discrete-time index.

Since the Fourier transform is linear, this relation is also additive in the frequency domain

$$Y[k] = \mathcal{F}\{y[n]\} = S[k] + D[k] \quad (2)$$

where $Y[k]$, $S[k]$ and $D[k]$ are the discrete Fourier transform (DFT) of $y[n]$, $s[n]$ and $d[n]$, respectively, and k is the discrete-frequency index. In practice, the spectral subtraction algorithm operates on a short-time signal, by dividing the noisy speech $y[n]$ into frames of size N . Then, the discrete Fourier transform is applied on each frame. The frequency resolution depends both on the sampling frequency f_s and the length of the frame in samples N . The discrete frequencies are defined by $f_k = kf_s/N$ for $0 \leq k < N/2$, and by $f_k = (k - N)f_s/N$ for $N/2 \leq k < N$. As a complex function, the Fourier transform of the corrupted signal can be represented by its rectangular form $Y[k] = Y_r[k] + jY_i[k]$, where $Y_r[k]$ and $Y_i[k]$ are the real and imaginary part of $Y[k]$, respectively. It can also be represented by its polar form $Y[k] = |Y[k]|e^{j\varphi_y[k]}$, where $|Y[k]|$ and $\varphi_y[k]$ are the magnitude and the phase of $Y[k]$, respectively.

2.1. Basic Spectral Subtraction

In the spectral subtraction method, the spectrum of the enhanced speech is obtained by subtracting an estimate of noise spectrum from the noisy signal spectrum [1]. To avoid negative magnitude spectrum, a simple half-wave rectifier has been first employed.

$$|\hat{S}[k]| = \max \left\{ |Y[k]| - |\hat{D}[k]|, 0 \right\} \quad (3)$$

where the noise spectrum, $\hat{D}[k]$, is estimated during the non-speech segments.

To reconstruct the enhanced signal, its phase $\varphi_s[k]$ is approximated by the phase $\varphi_y[k]$ of the noisy signal. This is based on the fact that in human perception the short time spectral magnitude is more important than the phase [4,20]. Thus, the discrete Fourier transform (DFT) of the enhanced speech is estimated as

$$\hat{S}[k] = |\hat{S}[k]| e^{j\varphi_y[k]} \quad (4)$$

Finally, the enhanced speech $\hat{s}[n]$ is obtained by inverse discrete Fourier transform (IDFT).

$$\hat{s}[n] = \mathcal{F}^{-1}\{\hat{S}[k]\} \quad (5)$$

However, this basic method suffers from a perceptually annoying residual noise named musical noise.

2.2. Generalized Spectral Subtraction

A generalized form of the spectral subtraction method has been suggested [2] to minimize the residual musical noise. It consists of over-subtracting an estimate of the noise power spectrum and ensuring that the resulting spectrum does not fall below a predefined minimum level (spectral floor).

$$|\hat{S}[k]|^\gamma = \max\left\{|Y[k]|^\gamma - \alpha|\hat{D}[k]|^\gamma, \beta|\hat{D}[k]|^\gamma\right\} \quad (6)$$

where $\alpha \geq 1$ is the over-subtraction multiplication factor and $0 < \beta \ll 1$ is the spectral flooring parameter [2]. γ is the exponent determining the transition sharpness, where $\gamma = 1$ corresponds to the magnitude spectral subtraction [1] and $\gamma = 2$ to the power spectral subtraction [2].

To minimize the speech distortion produced by large values of α , it has been proposed to let α vary from frame to frame within speech signal [2].

$$\alpha = \begin{cases} 4.75 & \text{SNR} \leq -5; \\ 4 - \frac{3}{20}\text{SNR} & -5 \leq \text{SNR} \leq 20; \\ 1 & \text{SNR} \geq 20. \end{cases} \quad (7)$$

where SNR is the segmental signal-to-noise ratio estimated in the frame and defined by:

$$\text{SNR} = 10 \log_{10} \left(\frac{\sum_{k=0}^{N-1} |Y[k]|^2}{\sum_{k=0}^{N-1} |\hat{D}[k]|^2} \right) \quad (8)$$

As for the basic method, the discrete Fourier transform of the enhanced signal $\hat{S}[k]$ is calculated from the estimated magnitude $|\hat{S}[k]|$ of the enhanced signal and the phase $\varphi_y[k]$ of the corrupted input signal using (4). The enhanced signal $\hat{s}[n]$ is reconstructed by inverse discrete Fourier transform (5).

2.3. Multi-Band Spectral Subtraction

The multi-band spectral subtraction method has been developed to reduce additive colored noise that does not uniformly affect the entire frequency band of the speech signal [3]. In this method, both noisy speech and estimated noise spectra are divided into M non-overlapping frequency bands. Then, the generalized spectral subtraction is applied independently in each band [3]. The power spectrum estimate of the enhanced speech in the i th frequency band is obtained as:

$$|\hat{S}_i[k]|^2 = \max\{|Y_i[k]|^2 - \alpha_i \delta_i |\hat{D}_i[k]|^2, \beta |\hat{D}_i[k]|^2\} \quad b_i \leq k \leq e_i \quad (9)$$

where b_i and e_i are the beginning and the ending frequency bins of the i th frequency band ($1 \leq i \leq M$), α_i is the over-subtraction factor of the i th frequency band, and δ_i is the tweaking factor of the i th frequency band [3]. The over-subtraction factor α_i is related to the segmental SNR _{i} of the i th frequency band by:

$$\alpha_i = \begin{cases} 4.75 & \text{SNR}_i \leq -5; \\ 4 - \frac{3}{20}\text{SNR}_i & -5 \leq \text{SNR}_i \leq 20; \\ 1 & \text{SNR}_i \geq 20. \end{cases} \quad (10)$$

The segmental SNR_{*i*} of the *i*th frequency band is defined by:

$$\text{SNR}_i = 10 \log_{10} \left(\frac{\sum_{k=b_i}^{e_i} |Y_i[k]|^2}{\sum_{k=b_i}^{e_i} |\hat{D}_i[k]|^2} \right) \quad (11)$$

where *b_i* and *e_i* are the beginning and the ending frequency bins of the *i*th frequency band. It can also be expressed using the natural logarithmic function:

$$\text{SNR}_i = \frac{10}{\ln(10)} \left(\ln \left(\sum_{k=b_i}^{e_i} |Y_i[k]|^2 \right) - \ln \left(\sum_{k=b_i}^{e_i} |\hat{D}_i[k]|^2 \right) \right) \quad (12)$$

The tweaking factor δ_{*i*} in (9) can be used to have an additional degree of noise removing control in each frequency band. The values of δ_{*i*} are experimentally defined and set to [3]:

$$\delta_i = \begin{cases} 1 & f_i \leq 1 \text{ kHz}; \\ 2.5 & 1 \text{ kHz} < f_i \leq 2 \text{ kHz}; \\ 1.5 & 2 \text{ kHz} < f_i \leq 4 \text{ kHz}. \end{cases} \quad (13)$$

where *f_i* denotes frequency in the *i*th band.

Figure 1 shows the block diagram of the speech enhancement system based on the multi-band spectral subtraction approach. Input noisy speech, *y*[*n*], is segmented into consecutive frames of *N* samples before applying discrete Fourier transform (DFT). The magnitude, |*Y*[*k*]>, and phase, φ_{*y*}[*k*], of the Fourier transformed signal are calculated. Then, spectrum of the noisy speech, |*Y*[*k*]², is calculated for the current frame, while the spectrum of the noise, | \hat{D} [*k*]², is estimated during non-speech segments. Both spectra are separated into (*M* = 4) frequency bands of 1 kHz in width each, for a sampling frequency *f_s* = 8 kHz. The segmental SNR_{*i*} and the over-subtraction factor α_{*i*} are calculated for each frequency band to allow independent spectral subtraction. The 4 separate spectra of the enhanced speech, | \hat{S}_i [*k*]², are then merged and square root calculated to obtain $\hat{S}[k]$. Finally, the enhanced signal $\hat{s}[n]$ is reconstructed by using inverse discrete Fourier transform (IDFT).

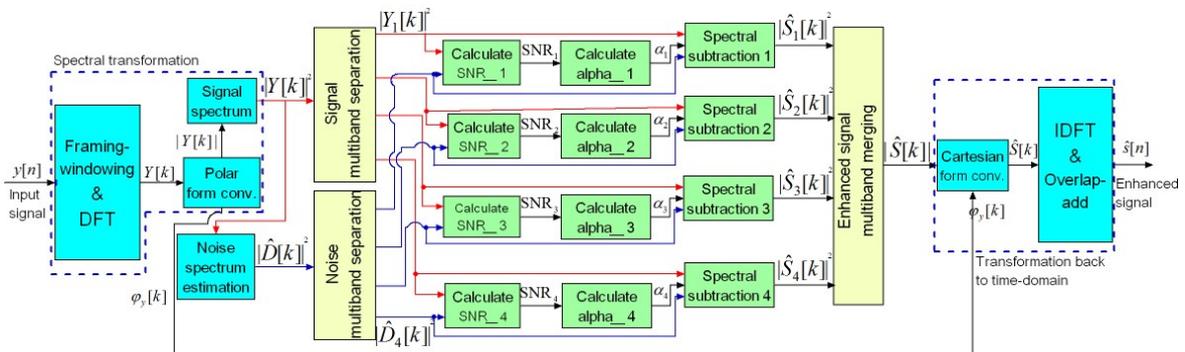


Figure 1. Block diagram of the multi-band spectral subtraction for speech enhancement. DFT: discrete Fourier transform; IDFT: inverse discrete Fourier transform.

3. FPGA Implementation

The proposed architecture has been implemented on a low-cost Artix-7 FPGA chip using a high-level programming tool (Xilinx System Generator), in MATLAB/SIMULINK (The Mathworks Inc., Natick, MA, USA) environment, and Nexys-4 development board. The top-level Simulink diagram of this architecture is presented in Figure 2, which principally corresponds to the block diagram presented in Figure 1. The proposed architecture uses some subsystems (blocks) developed in the past to implement, on an FPGA chip, the basic spectral subtraction method [15]. However, only subsystems related to the multi-band approach will be described in details in the following subsections:

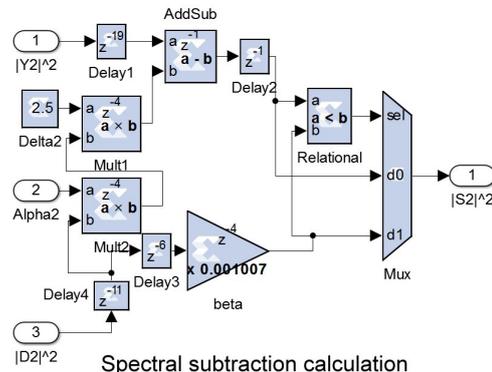
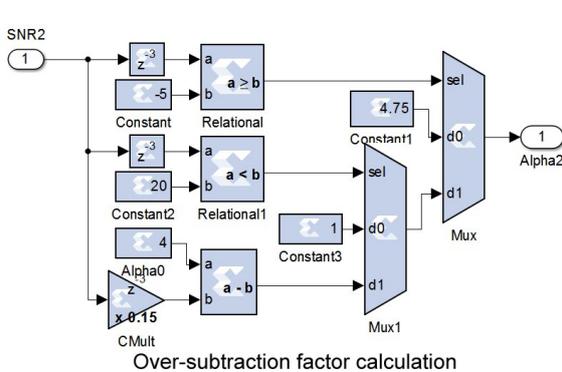
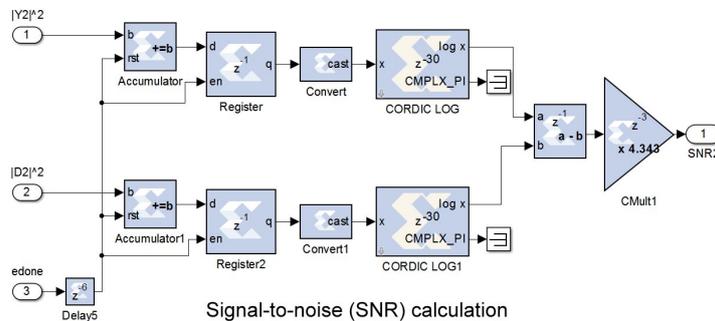
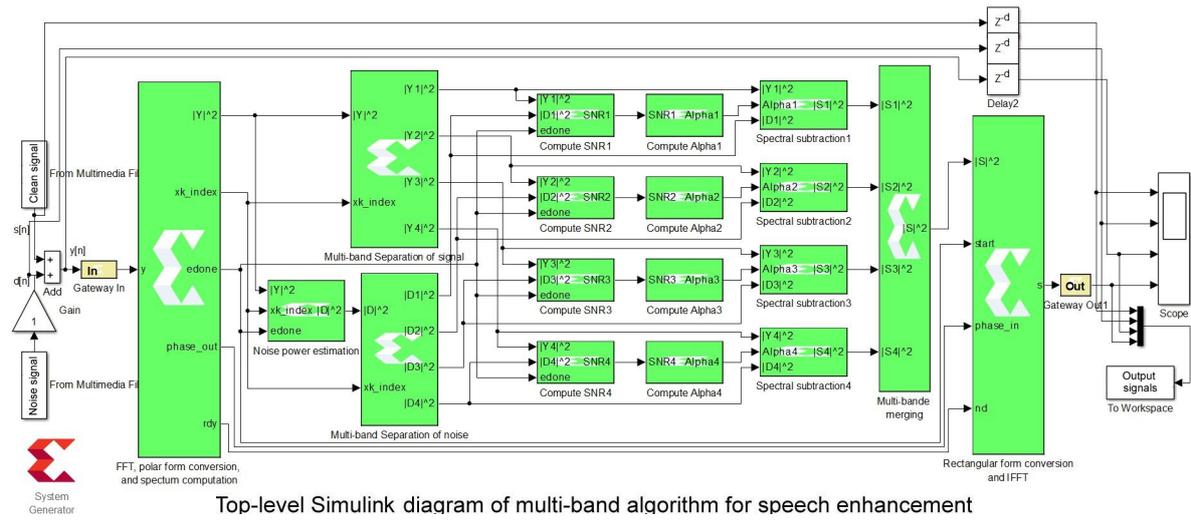


Figure 2. The proposed pipelined architecture of multi-band spectral subtraction using Xilinx system generator (XSG) blockset. The top-level Simulink diagram is shown on the top followed by details of the main calculation subsystems (signal-to-noise ratio, over-subtraction factor, and spectral subtraction). Details about the implementation of noise spectrum estimate can be found in [15].

3.1. Spectral Transformation and Noise Spectrum Estimation

This step corresponds to the four left blue blocks in Figure 1 and the two left XSG subsystems in Figure 2. As described in [15], the spectral analysis is performed by a Xilinx FFT (Fast Fourier Transform) block that provides the real, $Y_r[k]$, and imaginary, $Y_i[k]$, parts of the transformed signal $Y[k]$. It also provides the frequency index k output and an output *done* indicating that the computation of the Fourier transform of the current frame is complete and ready to output. Then, the Xilinx CORDIC (COordinate Rotation DIgital Computer) block is used to convert the transformed signal to its polar form, i.e., magnitude ($|Y[k]|$) and phase ($\varphi_y[k]$). A simple multiplier is used to calculate the power spectrum, $|Y[k]|^2$, of the noisy signal.

On the other hand, the power spectrum, $|\hat{D}[k]|^2$, of the additive noise is estimated using its average value calculated during the first five frames. A RAM (Random Access Memory)-based accumulator is used to estimate the noise power. More details on this subsystem can be found in [15].

3.2. Multi-Band Separation of Signal and Noise

This step corresponds to the two left yellow blocks in Figure 1 and their associated two subsystems in Figure 2. The hardware implementation of this subsystem is done using four register (one per frequency band) having as input the signal power spectrum, $|Y[k]|^2$, and driven by the frequency index signal, k . If k belongs to the i th frequency band, then the i th register is enabled by k , i.e., $|Y_i[k]|^2 = |Y[k]|^2$. Otherwise, it is reset, i.e., $|Y_i[k]|^2 = 0$. For the i th band, the frequency index k is delimited by b_i and e_i , as in (11) and (12). The same subsystem is used to separate the noise power spectrum into linearly separated multi-band.

3.3. Signal-To-Noise (SNR) Estimator

This step corresponds to the four left green blocks in Figure 1 and their associated four subsystems in Figure 2. Considering the fact that Xilinx CORDIC block can calculate only the natural logarithm (ln) and not the decimal logarithm (log) function, the SNR subsystem has been implemented using (12) instead of (11). In addition, this approach permits avoiding the use of divider.

As shown in Figure 2, this subsystem uses accumulators and registers to compute the sums for both signal and noise, followed by the CORDIC blocks to calculate their respective ln functions. After the subtractor block, the resulting value is multiplied by constant $4.3429 = 10 / \ln(10)$.

3.4. Over-Subtraction Factor Calculation

This step corresponds to the four middle green blocks in Figure 1 and their associated four subsystems in Figure 2. Based on (10), this subsystem is implemented using two multiplexers, two comparators, one subtractor, and constant blocks. The subsystem used in the 2nd frequency band is shown in Figure 2.

3.5. Spectral Subtraction

This step corresponds to the four right green blocks in Figure 1 and their associated four subsystems in Figure 2. This subsystem is implemented using one comparator, one multiplexer, and three multipliers, according to (9). Figure 2 shows the subsystem used in the 2nd frequency band.

3.6. Multi-Band Merging

This step corresponds to the right yellow blocks in Figure 1 and its associated subsystem in Figure 2. The subsystem is implemented using tree Xilinx adder blocks to merge the spectra of the four sub-bands.

3.7. Transformation Back to Time-Domain

This step corresponds to the two right blue blocks in Figure 1 and the right XSG subsystem in Figure 2. The Fourier transform of the enhanced signal $S[k]$ is first converted to the rectangular form (to real, $S_r[k]$, and imaginary, $S_i[k]$, parts) using a Xilinx CORDIC block, then transformed back to time domain ($s[k]$) using XSG IFFT (Inverse Fast Fourier Transform) block. More details on this subsystem can be found in [15].

3.8. Pipelining

The pipelining consists in reducing the critical path delay by inserting delays into the computational elements (multipliers and adders) to increase the operating frequency (Figure 2). The critical path corresponds to the longest computation time among all paths that contain zero delays [21]. About 410 delays have been inserted in different paths and balanced to ensure synchronization. The proposed pipelining increased substantially the operating frequency, but at the cost of the output latency of 36 samples. More details on how delays are inserted and redistributed can be found in our previous works [21,22].

3.9. Implementation Characteristics

Table 1 shows the hardware resources utilization, the maximum operating frequency, and the power consumption for the Artix-7 XC7A100T FPGA chip, as reported by Xilinx ISE 14.7 tool (Xilinx Inc, San Diego, CA, USA). The proposed pipelined architecture consumes 4955 logic slices from 15,850 available on this chip (31.2%). Also, it consumes 59 DSP48E1s from 240 available (24.6%). It occupies a small part of this low-cost FPGA. Therefore, the used resources consume about 107 mW. It can be noted that the pipelining of the implemented architecture increased the operating frequency from 24 MHz to 125 MHz, at the cost of an output latency of 36 delays. Therefore, the pipelined architecture requires more flip-flops (20,020 instead of 16,287) because of the inserted delays. Both default and pipelined architectures use the same number DSP48E1s and RAMB18E1s.

Table 1. Resource utilization, maximum operating frequency, and total power consumption obtained for the Artix-7 XC7A100T chip. LUT: Look-Up Table; IOB: Input/Output Block.

| Architecture | Default | Pipelined |
|-----------------------------|----------------|----------------|
| Resource utilization | | |
| Slices (15,850) | 4,617 (29.1%) | 4,955 (31.2%) |
| Flip Flops (126,800) | 16,287 (12.8%) | 20,020 (15.8%) |
| LUTs (63,400) | 15,067 (23.7%) | 16,541 (26.1%) |
| Bonded IOBs (210) | 42 (20.0%) | 42 (20.0%) |
| RAMB18E1s (270) | 6 (2.2%) | 6 (2.2%) |
| DSP48E1s (240) | 59 (24.6%) | 59 (24.6%) |
| Maximum Operating Frequency | 24.166 MHz | 125.094 MHz |
| Total power consumption | 107 mW | 107 mW |

It can be noted that a 32-bit fixed-point format with 28 fractional bits has been globally used to quantify data. However, a 24-bit fixed-point format with 18 fractional bits has been sufficient to compute the over-subtraction factor α_i and the segmental SNR $_i$.

3.10. Hardware/Software Co-Simulation

Software simulation of the designed XSG-based model provides a faithful behavior to that performed on hardware (bit and cycle accurate modeling). This allows us to take full advantage of the simulation environment of SIMULINK to visualize intermediate signals and facilitate the tuning of the XSG block parameters in order to reach the desired performance. It also optimizes resources

by choosing the number of bits needed to quantify data in different paths that ensures the needed performances. However, the designed XSG-based architecture can be executed on actual FPGA chip using the hardware-in-the-loop co-simulation from MATLAB/SIMULINK environment [23]. A number of development boards are pre-configured on the XSG tool, but the Nexys-4 board is not included and must be configured manually. This compilation mode generates a bitstream file and its associate gray SIMULINK block (Figure 3).

During the hardware/software co-simulation, the compiled model (bitstream file) is uploaded and executed on the FPGA chip from SIMULINK environment. XSG tool takes data from the input wav files in SIMULINK environment and transmits them to the design on the FPGA board using the JTAG (Joint Test Action Group) connection. It reads the enhanced signal (output) back from JTAG and sends it to SIMULINK for storage or display.

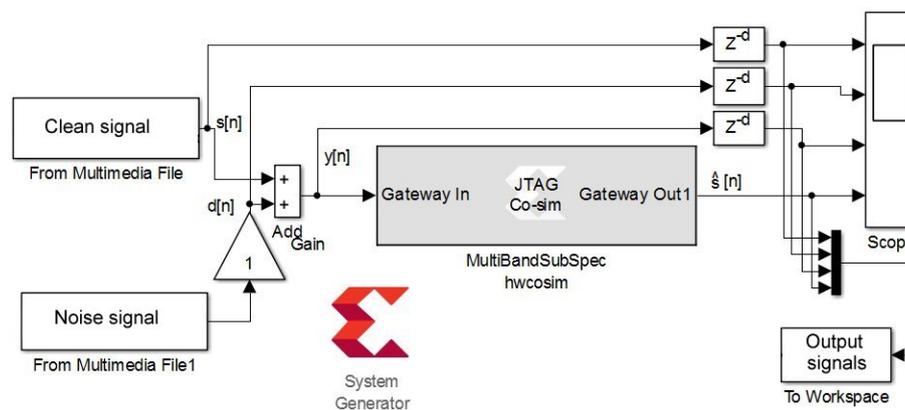


Figure 3. Hardware co-simulation corresponding to the diagram of Figure 2, where $s[n]$ is clean speech signal, $d[n]$ is the noise, $y[n]$ is the corrupted speech signal, and $\hat{s}[n]$ is the enhanced speech. JTAG: Joint Test Action Group protocol.

4. Results and Discussion

The proposed architecture has been tested on hardware using natural speech corrupted by artificial and actual additional noises, with sampling frequency of 8 kHz. The main objective of our experimental test is to validate the implementation process by ensuring that the hardware architecture (XSG) gives the same enhancement performances than the software simulation (MATLAB). Comparison of the multi-band spectral subtraction algorithm to other speech enhancement methods has been evaluated in the literature.

Figures 4 and 5 show the enhancement performances for speech corrupted by artificial blue and pink noises, respectively. However, Figures 6 and 7 present the enhancement performances for speech corrupted by actual car and jet noises, respectively. The fixed-point XSG implementation of the multi-band spectral subtraction technique performs as well as the floating-point MATLAB simulation. Waveforms and spectrograms of the speech signals enhanced by hardware and software are similar. The experimental tests prove the accuracy of the FPGA-based implementation. It can be noted that noise was estimated during the first five frames.

On the other hand, the spectrograms of Figures 4–7 show that the additive noises are removed despite the fact that they do not uniformly affect the frequency band of the speech. For each figure, the difference (error) between the signal enhanced by XSG-based architecture and the signal enhanced by MATLAB simulation has been represented with the same scale in the time-domain. Its time-frequency characteristics seem to be close to those of a white noise (quantization noise).

The enhancement performances of the proposed XSG-based architecture are also compared to those obtained by MATLAB simulation using two objective tests: the overall signal-to-noise ratio (oveSNR) and the segmental signal-to-noise ratio (segSNR).

The overall signal-to-noise ratio (oveSNR) of the enhanced speech signal is defined by:

$$\text{oveSNR}_{\text{dB}} = 10 \log_{10} \left(\frac{\sum_{n=0}^{L-1} s^2[n]}{\sum_{n=0}^{L-1} (\hat{s}[n] - s[n])^2} \right) \quad (14)$$

where $s[n]$ and $\hat{s}[n]$ are the original and enhanced speech signals, respectively, and L is the length of the entire signal in samples.

The segmental signal-to-noise ratio (segSNR) is calculated by averaging the frame-based SNRs over the signal.

$$\text{segSNR}_{\text{dB}} = \frac{1}{M} \sum_{m=0}^{M-1} 10 \log_{10} \left(\frac{\sum_{n=0}^{L_s-1} s^2[m, n]}{\sum_{n=0}^{L_s-1} (\hat{s}[m, n] - s[m, n])^2} \right) \quad (15)$$

where M is the number of frames, L_s is the frame size, and $s[m, n]$ and $\hat{s}[m, n]$ are the m -th frame of original and enhanced speech signals, respectively.

Table 2 presents the objective tests for noisy and enhanced speech signals. The results obtained by the proposed XSG-based architecture are approximately similar to those obtained by MATLAB simulation. The minor differences between oveSNR and segSNR can be explained by the quantification errors.

Table 2. Objective tests obtained for noisy and enhanced speech signals for various additive noises. XSG: Xilinx system generator; SNR: segmental signal-to-noise ratio

| Objective Test | Blue Noise | | | Pink Noise | | | Volvo Car Noise | | | F-16 Jet Noise | | |
|----------------|------------|--------|-------|------------|--------|-------|-----------------|--------|-------|----------------|--------|------|
| | Noisy | MATLAB | XSG | Noisy | MATLAB | XSG | Noisy | MATLAB | XSG | Noisy | MATLAB | XSG |
| oveSNR (dB) | 11.92 | 16.34 | 16.24 | 8.57 | 11.96 | 11.95 | 5.04 | 12.83 | 13.84 | 6.18 | 9.94 | 9.79 |
| SegSNR (dB) | 1.77 | 8.66 | 8.00 | -2.10 | 3.95 | 3.74 | -5.27 | 7.32 | 7.73 | -4.72 | 2.39 | 2.08 |

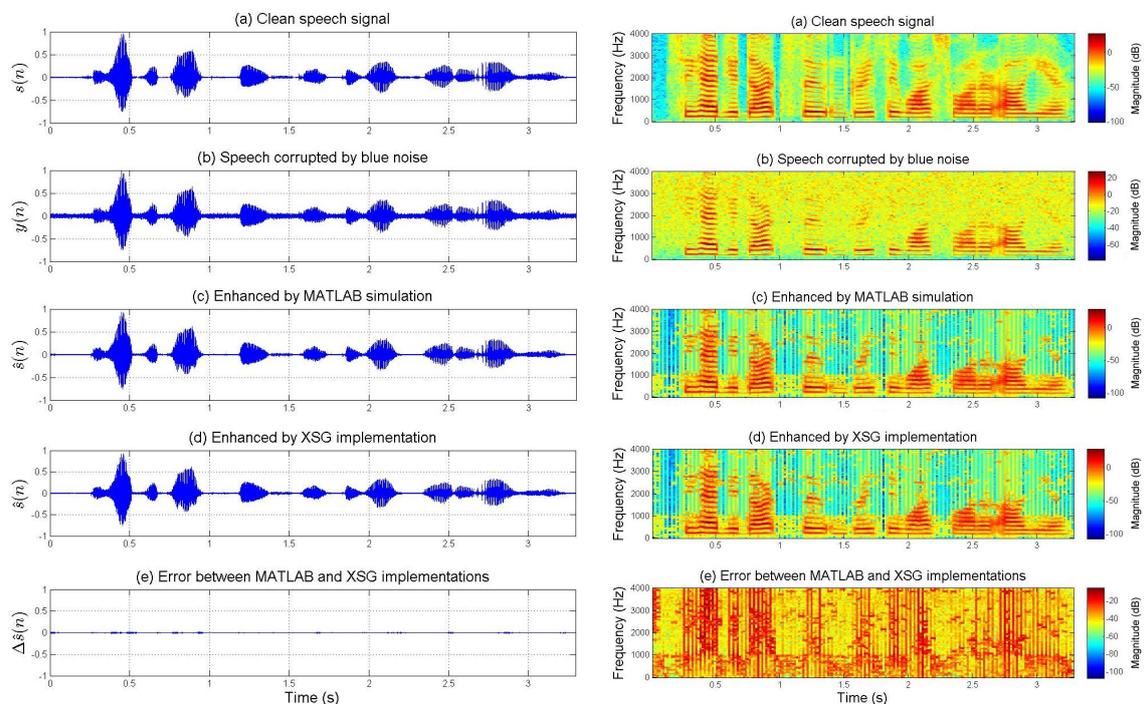


Figure 4. Time waveforms (left) and spectrograms (right) of clean speech signal (a), speech signal corrupted with artificial blue noise (b), enhanced speech with floating-point MATLAB simulation (c), enhanced with fixed-point XSG implementation (d), and error (difference) enhancement between MATLAB and XSG implementations (e).

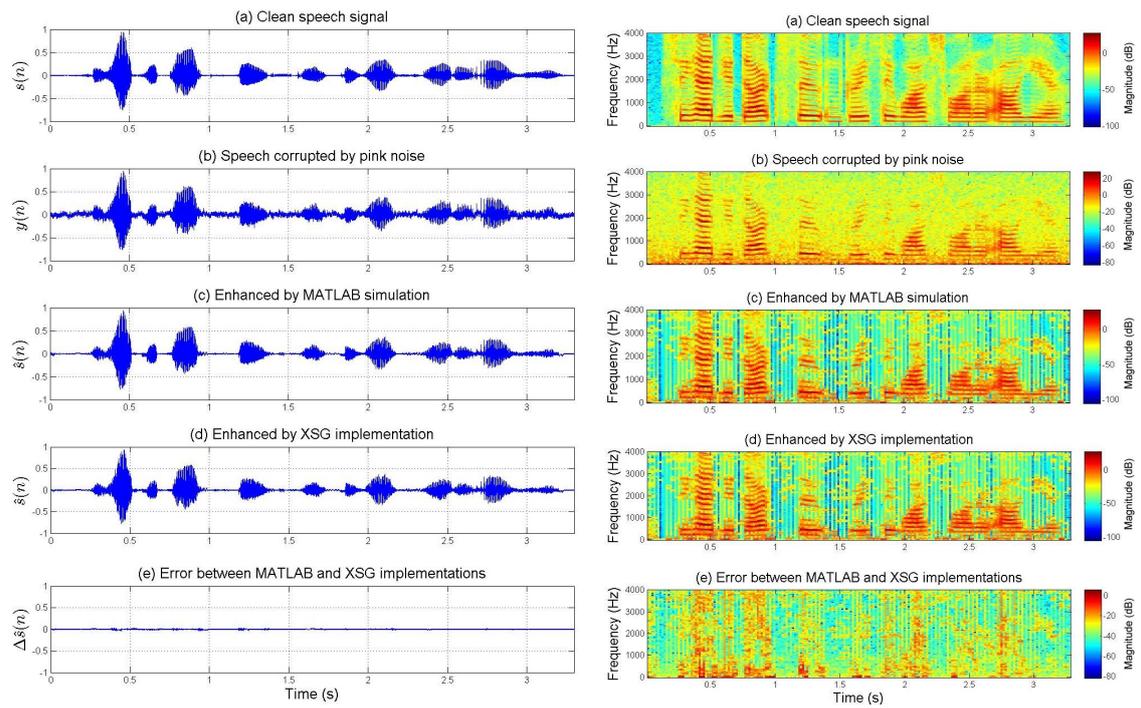


Figure 5. As in Figure 4 but for pink noise.

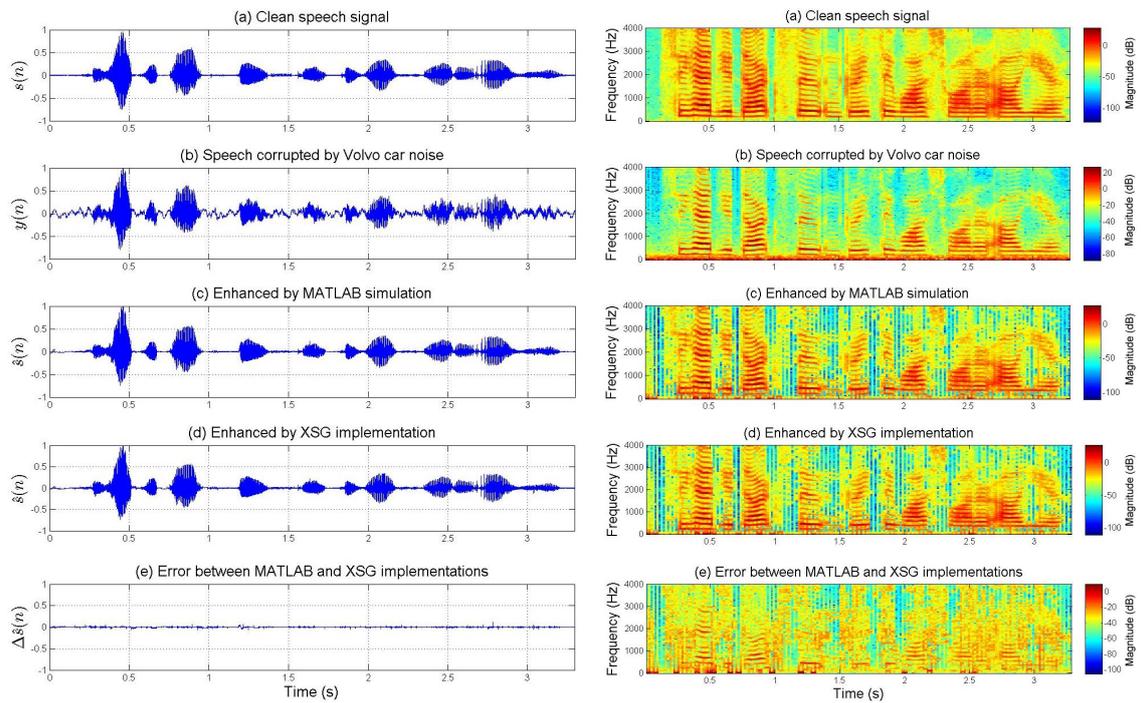


Figure 6. As in Figure 4 but for Volvo car noise.

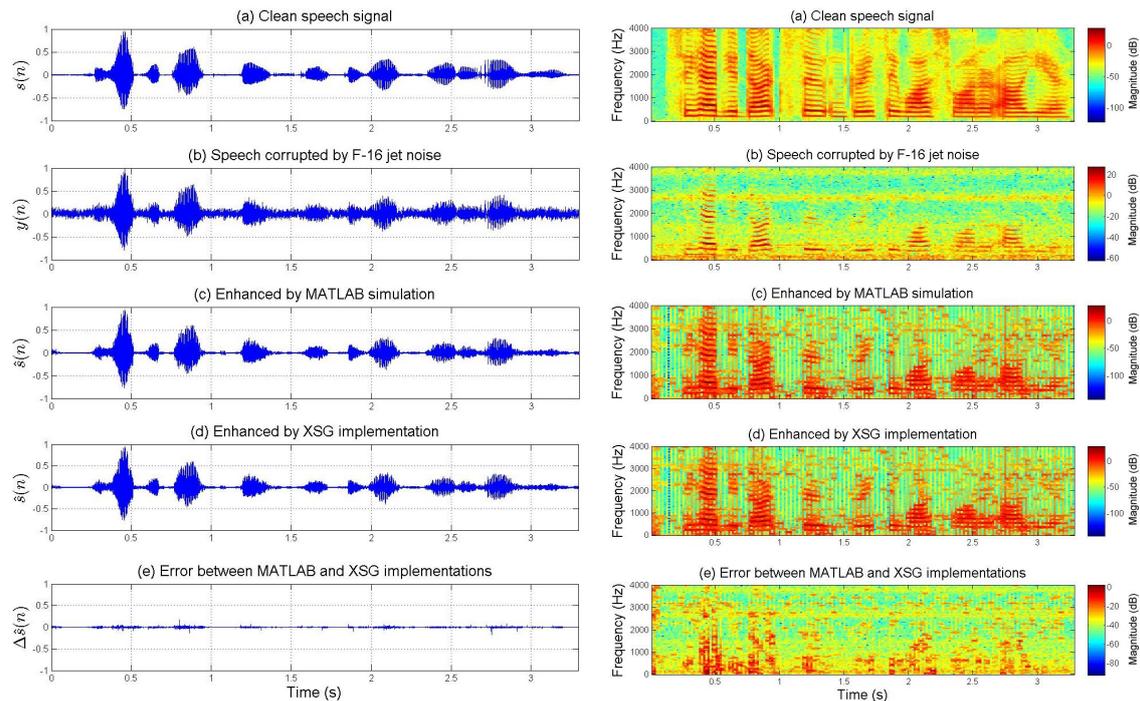


Figure 7. As in Figure 4 but for F-16 jet noise.

5. Conclusions

A pipelined architecture of multi-band spectral subtraction for speech enhancement has been implemented on a low-cost FPGA chip. It occupies a small part of the hardware resources (about 30% of the logic slices) and consumes a low power of 107 mW. The proposed pipelined architecture is five times faster than the default implementation (operating frequency of 125 MHz rather than 24 MHz). Performances obtained by the proposed architecture on speech corrupted by artificial and actual noise are similar to those obtained by MATLAB simulation. For each frequency band, the over-subtraction parameter is adjusted by the signal-to-noise ratio (SNR) to preserve the low-power speech components. However, in the future, a voice activity detector can be added to this architecture in order to update continuously the noise spectrum estimate.

Acknowledgments: This research is financially supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

Conflicts of Interest: The author declare no conflict of interest.

References

1. Boll, S.F. Suppression of Acoustic Noise in Speech Using Spectral Subtraction. *IEEE Trans. Acoust. Speech Signal Process.* **1979**, *27*, 113–120.
2. Berouti, M.; Schwartz, R.; Makhoul, J. Enhancement of speech corrupted by acoustic noise. In Proceedings of the IEEE International Conference on ICASSP 1979 Acoustics, Speech, and Signal Processing, Washington, DC, USA, 2–4 April 1979; Volume 4, pp. 208–211.
3. Kamath, S.; Loizou, P. A multi-band spectral subtraction method for enhancing speech corrupted by colored noise. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Orlando, FL, USA, 13–17 May 2002; Volume 4, pp. 4160–4164.
4. Udrea, R.M.; Vizireanu, N.; Ciochina, S.; Halunga, S. Nonlinear spectral subtraction method for colored noise reduction using multi-band Bark scale. *Signal Process.* **2008**, *88*, 1299–1303.
5. Upadhyay, N.; Karmakar, A. An Improved Multi-Band Spectral Subtraction Algorithm for Enhancing Speech in Various Noise Environments. *Procedia Eng.* **2013**, *64*, 312–321.

6. Simard, Y.; Bahoura, M.; Roy, N. Acoustic Detection and Localization of whales in Bay of Fundy and St. Lawrence Estuary Critical Habitats. *Can. Acoust.* **2004**, *32*, 107–116.
7. Dron, J.; Bolaers, F.; Rasolofondraibe, I. Improvement of the sensitivity of the scalar indicators (crest factor, kurtosis) using a de-noising method by spectral subtraction: Application to the detection of defects in ball bearings. *J. Sound Vib.* **2004**, *270*, 61–73.
8. Bouchikhi, E.H.E.; Choqueuse, V.; Benbouzid, M.E.H. Current Frequency Spectral Subtraction and Its Contribution to Induction Machines' Bearings Condition Monitoring. *IEEE Trans. Energy Convers.* **2013**, *28*, 135–144.
9. Yang, L.P.; Fu, Q.J. Spectral subtraction-based speech enhancement for cochlear implant patients in background noise. *Acoust. Soc. Am. J.* **2005**, *117*, 1001–1004.
10. Karunajeewa, A.; Abeyratne, U.; Hukins, C. Silence-breathing-snore classification from snore-related sounds. *Physiol. Meas.* **2008**, *29*, 227–243.
11. Chang, G.C.; Lai, Y.F. Performance evaluation and enhancement of lung sound recognition system in two real noisy environments. *Comput. Methods Programs Biomed.* **2010**, *97*, 141–150.
12. Emmanouilidou, D.; McCollum, E.D.; Park, D.E.; Elhilali, M. Adaptive Noise Suppression of Pediatric Lung Auscultations With Real Applications to Noisy Clinical Settings in Developing Countries. *IEEE Trans. Biomed. Eng.* **2015**, *62*, 2279–2288.
13. Whittington, J.; Deo, K.; Kleinschmidt, T.; Mason, M. FPGA implementation of spectral subtraction for in-car speech enhancement and recognition. In Proceedings of the 2nd International Conference on Signal Processing and Communication Systems, ICSPCS 2008, Gold Coast, Australia, 15–17 December 2008; pp. 1–8.
14. Mahbub, U.; Rahman, T.; Rashid, A.B.M.H. FPGA implementation of real time acoustic noise suppression by spectral subtraction using dynamic moving average method. In Proceedings of the IEEE Symposium on Industrial Electronics and Applications, ISIEA 2009, Kuala Lumpur, Malaysia, 4–6 October 2009; Volume 1, pp. 365–370.
15. Bahoura, M.; Ezzaidi, H. Implementation of spectral subtraction method on FPGA using high-level programming tool. In Proceedings of the 24th International Conference on Microelectronics (ICM), Algiers, Algeria, 16–20 December 2012; pp. 1–4.
16. Adiono, T.; Purwita, A.; Haryadi, R.; Mareta, R.; Priandana, E. A hardware-software co-design for a real-time spectral subtraction based noise cancellation system. In Proceedings of the 2013 International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS), Naha, Japan, 12–15 November 2013; pp. 5–10.
17. Kasim, M.; Adiono, T.; Fahreza, M.; Zakiy, M. Real-time Architecture and FPGA Implementation of Adaptive General Spectral Subtraction Method. *Procedia Technol.* **2013**, *11*, 191–198.
18. Oukherfellah, M.; Bahoura, M. FPGA implementation of voice activity detector for efficient speech enhancement. In Proceedings of the IEEE 12th International New Circuits and Systems Conference, Trois-Rivieres, QC, Canada, 22–25 June 2014; pp. 301–304.
19. Amornwongpeeti, S.; Ono, N.; Ekpanyapong, M. Design of FPGA-based rapid prototype spectral subtraction for hands-free speech applications. In Proceedings of the 2014 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), Siem Reap, Cambodia, 19–22 December 2014; pp. 1–6.
20. Wang, D.; Lim, J. The unimportance of phase in speech enhancement. *IEEE Trans. Acoust. Speech Signal Process.* **1982**, *30*, 679–681.
21. Bahoura, M.; Ezzaidi, H. FPGA-Implementation of Parallel and Sequential Architectures for Adaptive Noise Cancellation. *Circ. Syst. Signal Process.* **2011**, *30*, 1521–1548.
22. Bahoura, M. FPGA implementation of high-speed neural network for power amplifier behavioral modeling. *Analog Integr. Circ. Signal Process.* **2014**, *79*, 507–527.
23. Bahoura, M. FPGA Implementation of Blue Whale Calls Classifier Using High-Level Programming Tool. *Electronics* **2016**, *5*, 8, doi:10.3390/electronics5010008.

