Open Access
World Journal of Engineering and Technology, 2014, 2, 1-12
Published Online February 2014 in SciRes. http://www.scirp.org/journal/wjet
http://dx.doi.org/10.4236/wjet.2014.21001

Scientific Research

Wake-Up-Word Feature Extraction on FPGA

Veton Z. Këpuska, Mohamed M. Eljhani, Brian H. Hight

Electrical & Computer Engineering Department, Florida Institute of Technology, Melbourne, USA
Email: vkepuska@fit.edu, meljhani2009@my.fit.edu, bhight2008@my.fit.edu

Received 25 October 2013; revised 27 November 2013; accepted 5 December 2013

Copyright © 2014 Veton Z. Këpuska *et al*. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. In accordance of the Creative Commons Attribution License all Copyrights © 2014 are reserved for SCIRP and the owner of the intellectual property Veton Z. Këpuska *et al*. All Copyright © 2014 are guarded by law and by SCIRP as a guardian.

Abstract

Wake-Up-Word Speech Recognition task (WUW-SR) is a computationally very demand, particularly the stage of feature extraction which is decoded with corresponding Hidden Markov Models (HMMs) in the back-end stage of the WUW-SR. The state of the art WUW-SR system is based on three different sets of features: Mel-Frequency Cepstral Coefficients (MFCC), Linear Predictive Coding Coefficients (LPC), and Enhanced Mel-Frequency Cepstral Coefficients (ENH_MFCC). In (front-end of Wake-Up-Word Speech Recognition System Design on FPGA) [1], we presented an experimental FPGA design and implementation of a novel architecture of a real-time spectrogram extraction processor that generates MFCC, LPC, and ENH_MFCC spectrograms simultaneously. In this paper, the details of converting the three sets of spectrograms 1) Mel-Frequency Cepstral Coefficients (MFCC), 2) Linear Predictive Coding Coefficients (LPC), and 3) Enhanced Mel-Frequency Cepstral Coefficients (ENH_MFCC) to their equivalent features are presented. In the WUW-SR system, the recognizer's front-end is located at the terminal which is typically connected over a data network to remote back-end recognition (e.g., server). The WUW-SR is shown in Figure 1. The three sets of speech features are extracted at the front-end. These extracted features are then compressed and transmitted to the server via a dedicated channel, where subsequently they are decoded.

Keywords

Speech Recognition System; Feature Extraction; Mel-Frequency Cepstral Coefficients; Linear Predictive Coding Coefficients; Enhanced Mel-Frequency Cepstral Coefficients; Hidden Markov Models; Field-Programmable Gate Arrays

1. Introduction

In general, any automatic speech recognition system needs to be activated manually (push-to-talk), which needs

hand movement and hence mixed multi-modal interface. However, for people who use hands-busy applications, hand movement may be limited or impractical. This research work represents alternative solution to use Speech Only Interface. The solution that is being proposed is called Wake-Up-Word Speech Recognition (WUW-SR) [2]. A WUW-SR system would permit the user to control (stimulate) any hand-held device (Cell phone, Computer, etc.) with speech commands instead of hand engagements. In this paper we introduce three kinds of feature extraction of a new front-end model of the Wake-Up-Word Speech Recognition. We present an experimental design and implementation on FPGA of a novel architecture of a real-time feature extraction processor that generates three different features simultaneously. Our front-end can be added to any hand-held electronic device compatible with WUW-SR and control (trigger) it by using our voice only (no push to talk as is presently done). Our front-end is designed, simulated and implemented in Altera DSP development kit with Cyclone III FPGA as a portable system performing as a processor which is able to generate three different sets of features at a much faster rate than software. It is cost-effective, and consumes very little power, and it is not limited by having to operate on a general-purpose computer so it can be used on any portable device. The remainder of this paper is organized as follows: Section 2 provides feature extraction overview. Section 3 describes the Wake-Up-Word speech recognition architecture. Section 4 describes the front-end of WUW-SR design procedure and architecture. Section 5 describes the Mel-Frequency Cepstrum Coefficients (MFCC) algorithm. Section 6 describes the Autocorrelation Linear Predictive Coding (LPC) algorithm. Section 7 describes the Enhanced Mel-Frequency Cepstrum Coefficients (ENH-MFCC) algorithm. In Section 8 the results and comparisons of three spectrograms and features from FPGA hardware implementation are described and compared with the C++ front-end algorithm. These are followed by conclusions in Section 9.

2. Feature Extraction

The feature extraction of speech is one of the most important issues in the field of speech recognition. There are two dominant acoustic measurements of speech signal. One is the parametric modeling approach, which is developed to match closely the resonant structure of the human vocal tract that produces the corresponding speech sound. It is mainly derived from Linear Predictive analysis, such as LPC-based cepstrum (LPCC). The other approach is the nonparametric modeling method that is basically originated from the human auditory perception system. Mel-frequency cepstral coefficients (MFCCs) are utilized for this purpose [3]. In recent studies of speech recognition system, the MFCC parameters perform better than others in the recognition accuracy [4,5]. This paper presents the feature extraction solution based on MFCC, LPC and new set of features named Enhanced Mel-frequency Cepstral Coefficients (ENH-MFCC) with the architecture specially optimized for implementation in FPGA structures.

3. WUW-SR System Architecture

As shown in **Figure 1**, the WUW-SR can be broken down into three components as explained in (V. Z. Këpuska and T. B. Klein) [2]. The front-end system process takes an input pressure waveform (audio signal) and output a sequence of characteristic parameters MFCCs, LPCs, and ENH-MFCCs features. Whereas the back-end process takes this sequence and outputs the recognized command.

The audio signal processing module accepts raw audio samples and produces spectral representations of short time signals. The feature-extraction module generates features from this spectral representation, which are decoded with the corresponding hidden Markov's models (HMMs). The individual feature scores are classified using support vector machines (SVMs) into INV, OOV: in-, out-of-vocabulary speech.

4. Front-End of WUW-SR System Architecture

As shown in **Figure 2**, the design is divided into twenty four-modules (four-stages). The first seven pink-colored modules represent the pre-processing stage and are used as the basic modules to provide windowed speech signal to the other stages.

4.1. Stage A: Pre-Processing

1) Analog to Digital Converter ADC.

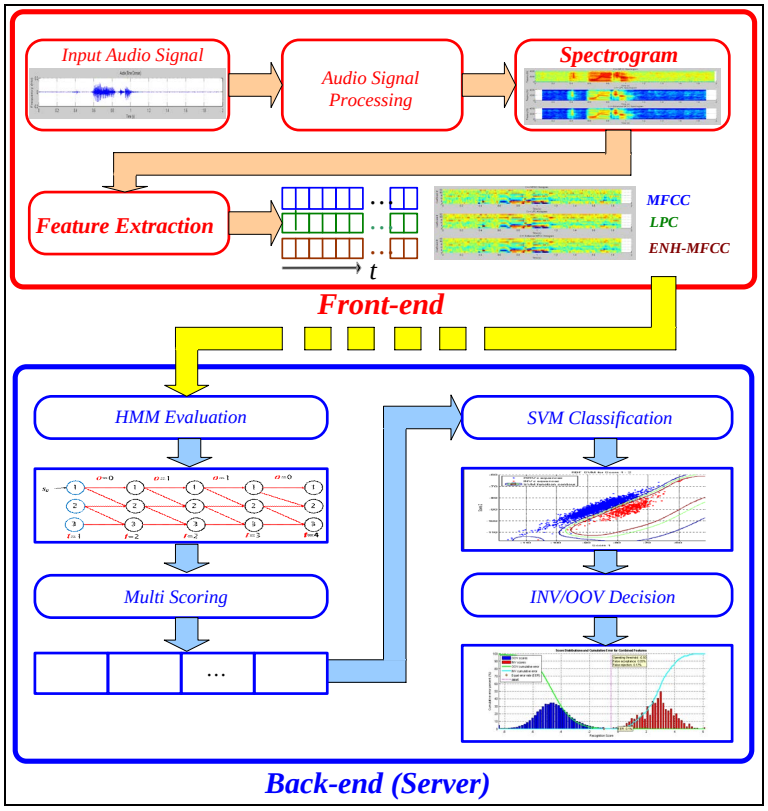

Figure 1. Overall WUW-SR architecture.

2) DC Filtering.
3) Serial to 32-Bit Parallel Converter.
4) Integer to Floating-Point Converter.
5) Pre-Emphasis Filtering.
6) Window Advance Buffering.
7) Hamming Window.

4.2. Stage B: Linear Predictive Coding Coefficients

The six yellow-colored modules represent the Linear Predictive Coding Coefficients (LPC) stage and are used to generate 12-Linear Predictive Coding features.

1) Autocorrelation Linear Predictive Coding.
2) Fast Fourier Transform FFT.
3) LPC Spectrogram.
4) Mel-Scale Filtering.
5) Discrete Cosine Transform DCT.
6) LPC Feature.

4.3. Stage C: Mel-Frequency Cepstral Coefficients

The five brown-colored modules represent the MFCC stage and are used to generate 12-MFCCs features.

1) Fast Fourier Transform FFT.
2) MFCC Spectrogram.
3) Mel-Scale Filtering.
4) Discrete Cosine Transform DCT.
5) MFCC Feature.

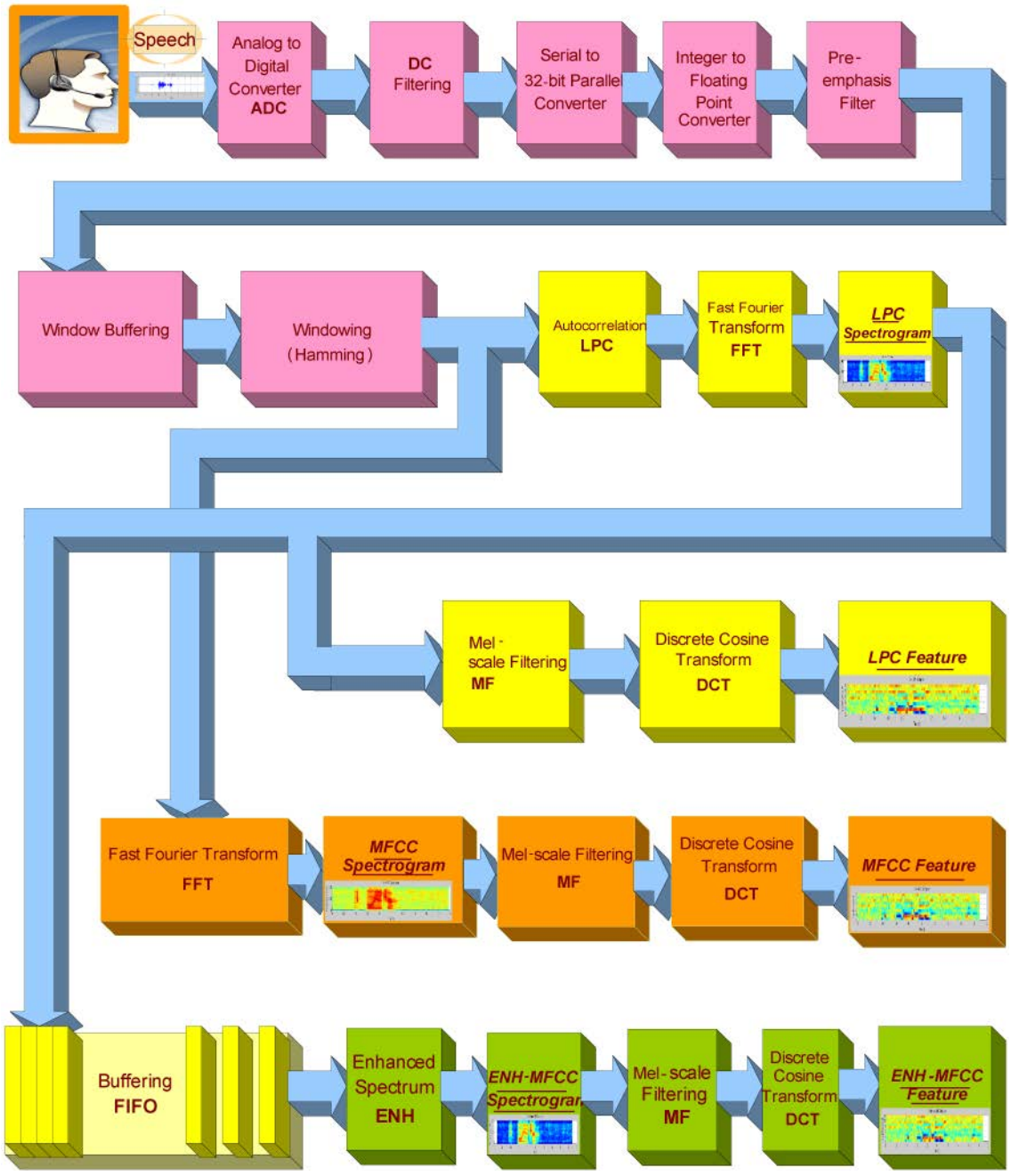

Figure 2. Overall front-end of WUW-SR block diagram.

4.4. Stage D: Enhanced Mel-Frequency Cepstral Coefficients

The five green-colored modules represent the ENH-MFCC stage and are used to generate 12-Enhanced MFCC features.

1) Enhanced Spectrum (ENH).
2) Enhanced MFCC Spectrogram.
3) Mel-Scale Filtering.
4) Discrete Cosine Transform DCT.
5) ENH-MFCC Feature.

5. Mel-Scale Frequency Cepstral Coefficients (MFCC) Feature Extraction

The feature extraction involves identifying the *formants* in the speech, which represent the frequency locations of energy concentrations in the speaker's vocal tract. There are many different approaches used: Mel-scale Frequency Cepstral Coefficients (MFCC), Linear Predictive Coding (LPC), Linear Prediction Cepstral Coefficients (LPCC), Reflection Coefficients (RCs). Among these, MFCC has been found to be more robust in the presence of background noise compared to other algorithms [6]. Also, it offers the best trade-offs between performance and size (memory) requirements. The primary reason for effectiveness of MFCC is that, it models the non-linear auditory response of the human ear which resolves frequencies on a log scale [7].

Intensive efforts have been carried out to achieve a high performance front-end. Converting a speech waveform into a form suitable for processing by the decoder requires several stages as shown in **Figure 3**:

1) Filtration: The waveform is sent through a low pass filter, typically 4 kHz to 8 kHz. As is evidenced by the bandwidth of the telephone system being around 4 kHz; this is sufficient for comprehension and used a minimum bandwidth required for telephony transmittal.

2) Analog-To-Digital Conversion: The process of digitizing and quantizing an analog speech waveform begin with this stage. Recall that the first step in processing speech is to convert the analog representations (first air pressure, and then analog electric signals from a microphone), into a digital signal.

3) Sampling Rate: The resulting waveform is sampled. Sampling rate theory requires a sampling (Nyquist) rate of double the maximum frequency (so 8 to 16 kHz as appropriate). The sampling rate of 8 kHz was used in our front-end (we used CODEC Chip to perform first, second, and third stages).

4) Serial to Parallel Converter: This model gets serial digital signal from CODEC and converts it to 32-bit.

5) Integer to Floating-Point Converter: This module converts 32-bit, signed integer data to single-precision (32-bit) floating-point values. The input data is routed through the int_2_float Mega function core named ALTFP_CONVERT.

6) Pre-Emphasis: The digitalized speech signal s(n) is put through a low-order LPF to spectrally flatten the signal and to make it less susceptible to finite precision effects later in the signal processing. The filter is represented by:

$$y[n] = x[n] - \alpha x[n-1].$$

$$\text{Output} = \text{Input} - (\text{PRE\_EMPH\_FACTOR} * \text{Previous\_input})$$

where we have chosen the value of

PRE_EMPH_FACTOR **(α)** as 0.975.

7) Window Buffering: A 32-bit, 256 deep dual-port RAM (DPRAM) stores 256 input samples. A state machine handles moving audio data into the RAM, and pulling data out of the RAM (40 samples) to be multiplied by the Hamming coefficients, which are stored in a ROM memory.

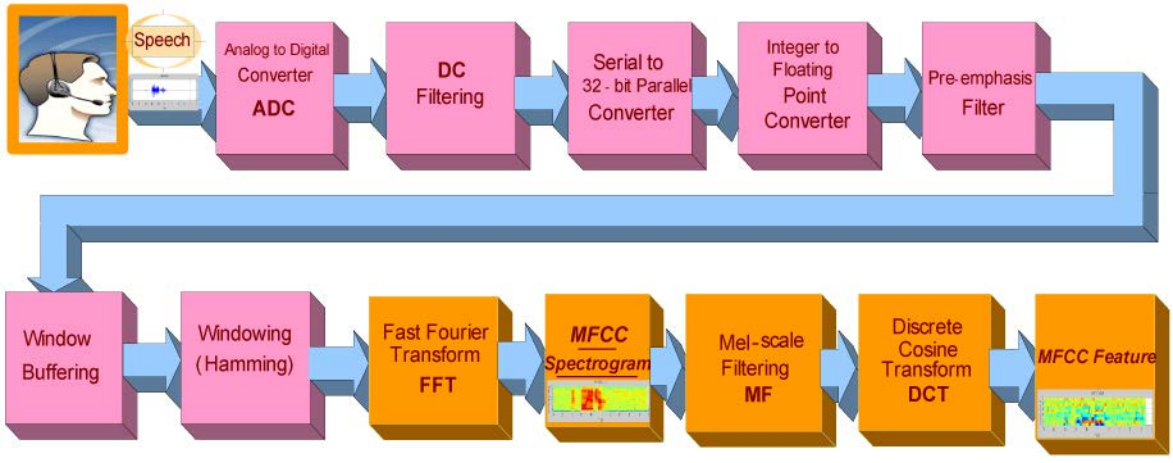

Figure 3. MFCC feature extraction.

8) Windowing: The Hamming window function smoothes the input audio data with a Hamming curve prior to the FFT function. This stage slices the input signal into discrete time segments. This is done by using window typically 25 ms wide (200 samples). A Hamming window size of 25 ms which consists of 200 samples at 8 KHz sampling frequency and 5 ms frame shift (40 samples) is picked for our front-end windowing.

9) Fast Fourier Transform: In order to map the sound data from the time domain to the frequency domain, the Altera IP Megafunction FFT module is used. The module is configured so as to produce a 256-point FFT. This function is capable of taking a streaming data input in natural order, and it can also output the transformed data in natural order, with maximum latency of 256 clock cycles once all the data (256 data samples) has been received.

10) Spectrogram: This module takes the complex data generated by the FFT and performs the function: **20*log10 (fft_real^2 + fft_imag^2).** We designed spectrogram to show how the spectral density of a signal varies with time. We used spectrogram module to identify phonetic sounds. Digitally sampled data, in the time domain, are broken up into chunks, which usually overlap, and Fourier transformed to calculate the magnitude of the frequency spectrum for each chunk. Each chunk then corresponds to a vertical line in the image; a measurement of magnitude versus frequency for a specific moment in time. The spectrums or time plots are then "laid side by side" to form the image surface.

11) Mel-Scale Filtering: While the resulting spectrum of the FFT contains information in each frequency in linear scale, human hearing is less sensitive at frequencies above 1000 Hz. This concept also has a direct effect on performance of ASR systems; therefore, the spectrum is warped using a logarithmic Mel scale. In order to create this effect on the FFT spectrum, a bank of filters is constructed with filters distributed equally below 1000 Hz and spaced logarithmically above 1000 Hz.

12) Discrete Cosine Transform: DCT is a Fourier-related transform similar to the discrete Fourier transform (DFT), but using only real numbers. DCTs are equivalent to DFTs of roughly twice the length, operating on real data with even symmetry (since the Fourier transform of a real and even function is real and even). A DCT computes a sequence of data points in terms of summation of cosine functions oscillating at various frequencies. The idea of performing DCT on Mel Scale is motivated by extraction of the speech frequency domain characteristics. DCT module reduces the speech signal's redundant information, and reaches the aim of regulating the speech signal into feature coefficients with minimal dimensions.

6. Autocorrelation Linear Predictive Coding (LPC) Feature Extraction

As shown in **Figure 4**, an additional module named Autocorrelation Linear Productive Coding (LPC) used to extract the speech as LPC features. The basic idea of LPC is to approximate the current speech sample as a linear combination of past samples as shown in the following equation:

$$x[n] = \sum_{k=1}^{p} a_k x[n-k] + e[n]$$

$x[n-k]$: Previous speech samples; p: Order of the model; a_k: Prediction coefficient; $e[n]$: Prediction error.

This module gets windowed data from the window module for representing the spectral envelope of a digital signal of speech in compressed form, using the information of a linear predictive model. We use this method to encode good quality speech and provide an estimate of speech parameters.

The goal of this method is to calculate prediction coefficients a_k for each frame. The order of LPC, which is the number of coefficients p, determines how closely the prediction coefficients can approximate the original spectrum. As the order increases, the accuracy of LPC also increases. This means the distortion will decrease. The main advantage of LPC is usually attributed to the all-pole characteristics of vowel spectra. Also, the ear is also more sensitive to spectral poles than zeros (M. R. Schroeder) [8]. In comparison to non-parametric spectral modeling techniques such as filter banks, LPC is more powerful in compressing the spectral information into few filter coefficients (K. K. Paliwal and W. B. Kleijn) [9].

7. Enhanced Mel-Scale Frequency Cepstral Coefficients (ENH-MFCC) Feature Extraction

The spectrum enhancement module is used to generate ENH-MFCC set of features. We have implemented this

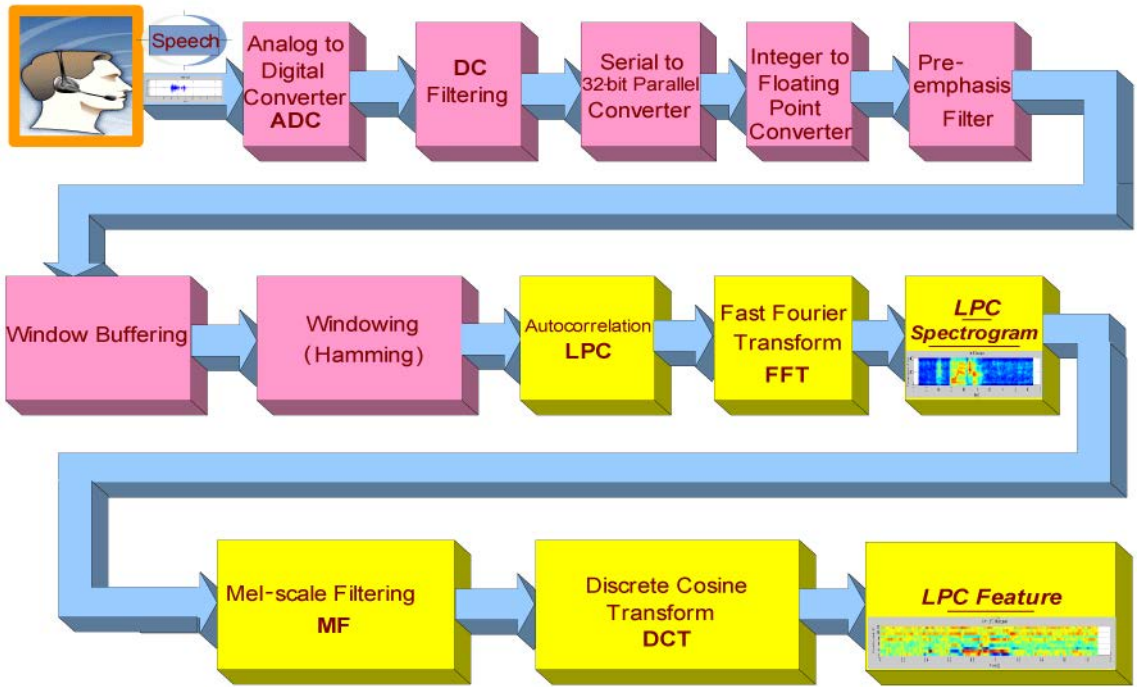

Figure 4. LPC feature extraction.

module as shown in the **Figure 5**, to perform an enhancement algorithm on the LPC spectrum signal. The ENH-MFCC features have a higher dynamic range than regular MFCC features, so these new features will help the back-end in improving the recognition quality and accuracy [2].

8. Results and Comparisons

In (front-end of Wake-Up-Word Speech Recognition System Design on FPGA) [1], we compared our spectrograms results with the (C, C++) WUW's front-end algorithm. We presented identical results from Hardware, and Software (C++) front-end. Because Wake-Up-Word Speech Recognition is a new concept, it is difficult to compare its front-end processor performance with existing front-ends. In order to perform a fair analysis we tested the performance of this system by comparing its three sets of spectrograms and features (MFCC, LPC, and ENH-MFCC) with the software (C, C++) WUW's front-end algorithm implementation. The front-end processor described in this paper has been modeled in Verilog HDL and implemented in low cost, high speed, and power efficient (Cyclone III EP3C120F780C7) FPGA on DSP development kit. The development of the front-end was conducted piecewise based on the modularity of the original software (C, C++) algorithm implementation and based on equivalent floating-point MATLAB implementation. Each module was tested after it was completed to ensure correct operation before the next block was developed. For example, the word "Voyager" with 8 KHz sampling rate was chosen as input audio data for testing our front-end. Testing was conducted by comparing spectrograms of triple features. In addition the produced triple features (MFCC, LPC, and ENH-MFCC) out of the hardware front-end model were compared with the software (C, C++) front-end. The results show:

1) As shown in **Figures 6**, **7** the MFCC, LPC, and ENH-MFCC spectrograms generated from Hardware and Software (C++) are identical.

2) In **Figures 8**, **9** the MFCC, LPC, and ENH-MFCC 12-features histograms generated from Hardware and Software (C++) are also identical.

3) We regenerated new MFCC, LPC, and ENH-MFCC histograms with 11-features by removing the first feature slice because of large dynamic range of the first feature that would make the remaining output features very small.

As shown in **Figures 10**, **11** the MFCC, LPC, and ENH-MFCC 11-features histograms generated from Hard-

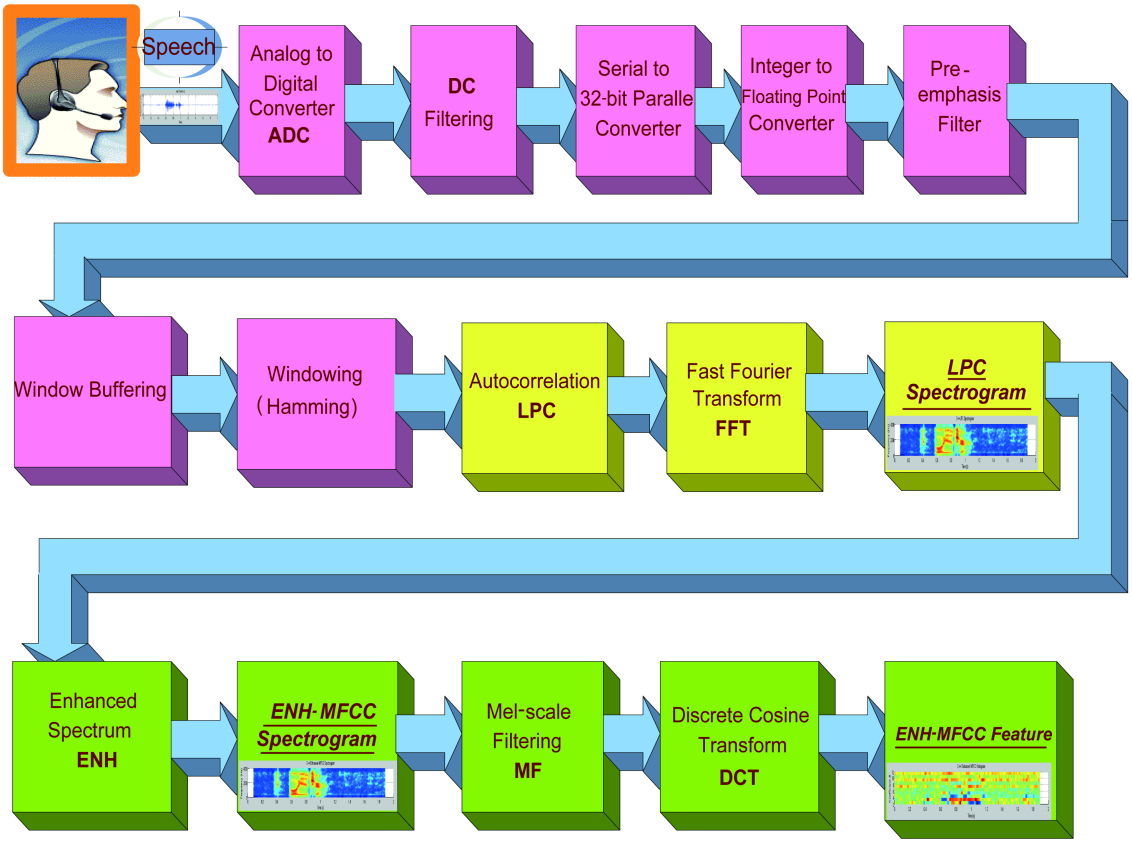

Figure 5. ENH-MFCC feature extraction.

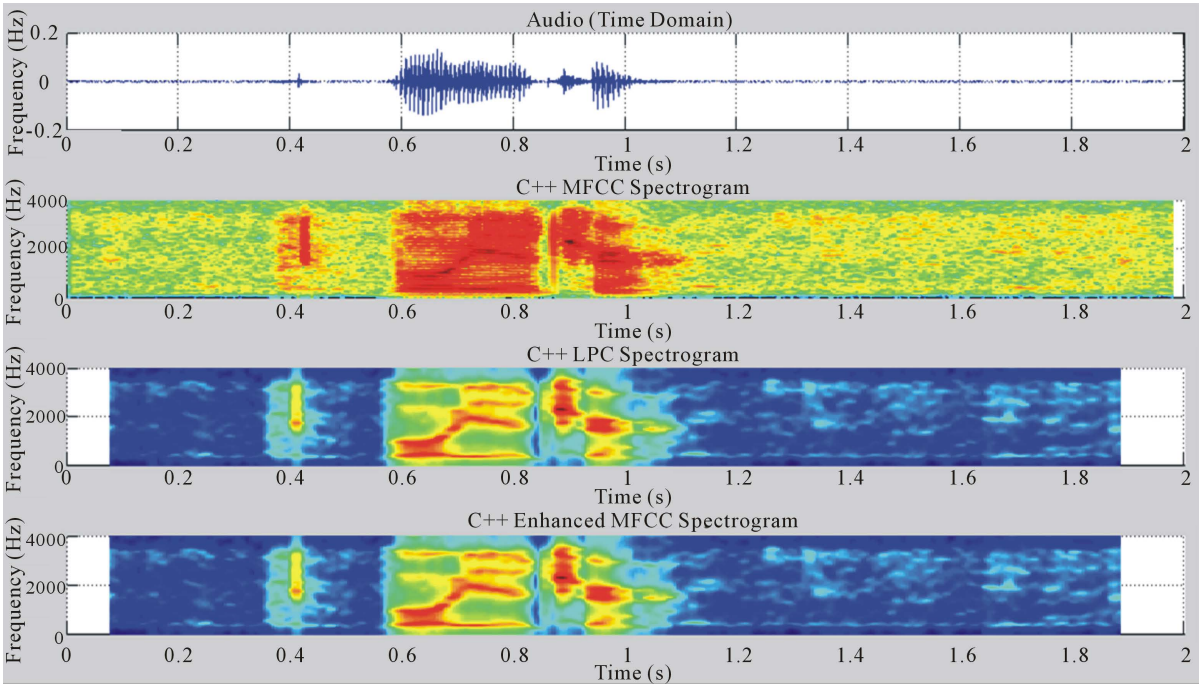

Figure 6. C++ frontend spectrograms with audio data.

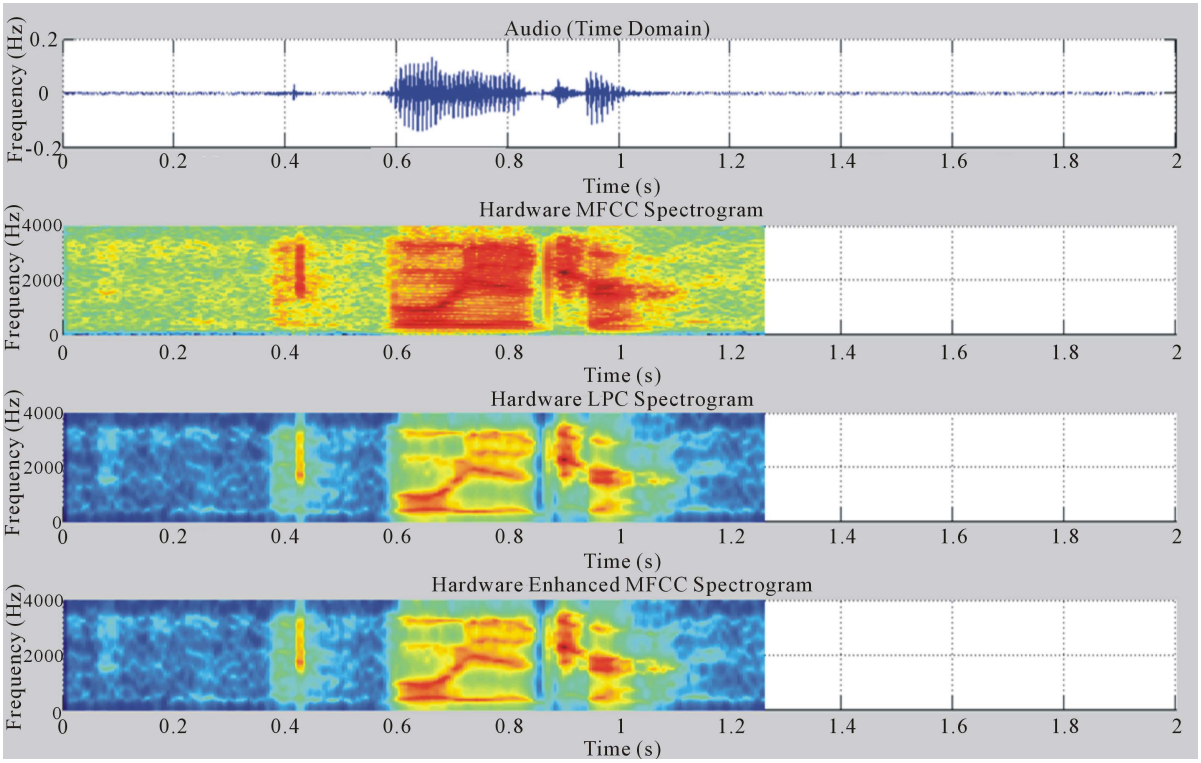

Figure 7. Hardware frontend spectrograms with audio data (due to limited amount of hardware resources the part of the data is not show in the resulting spectrograms).

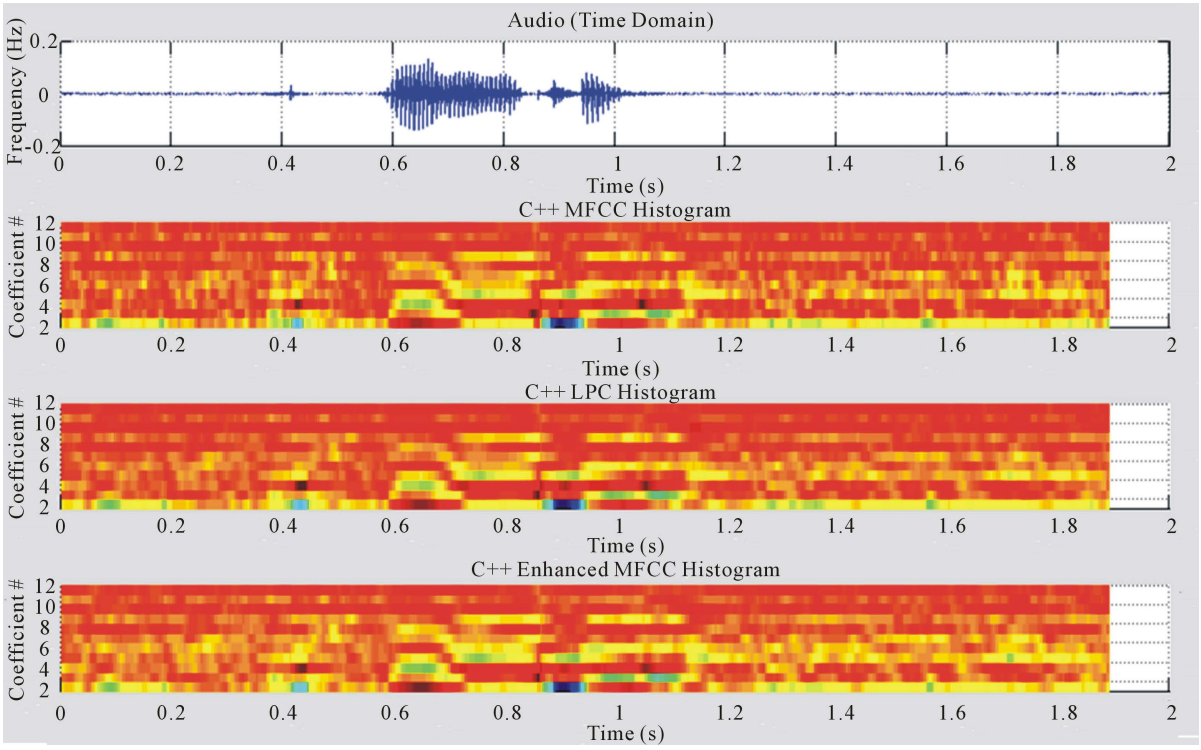

Figure 8. C++ frontend histograms with audio data (12_Coefficients).

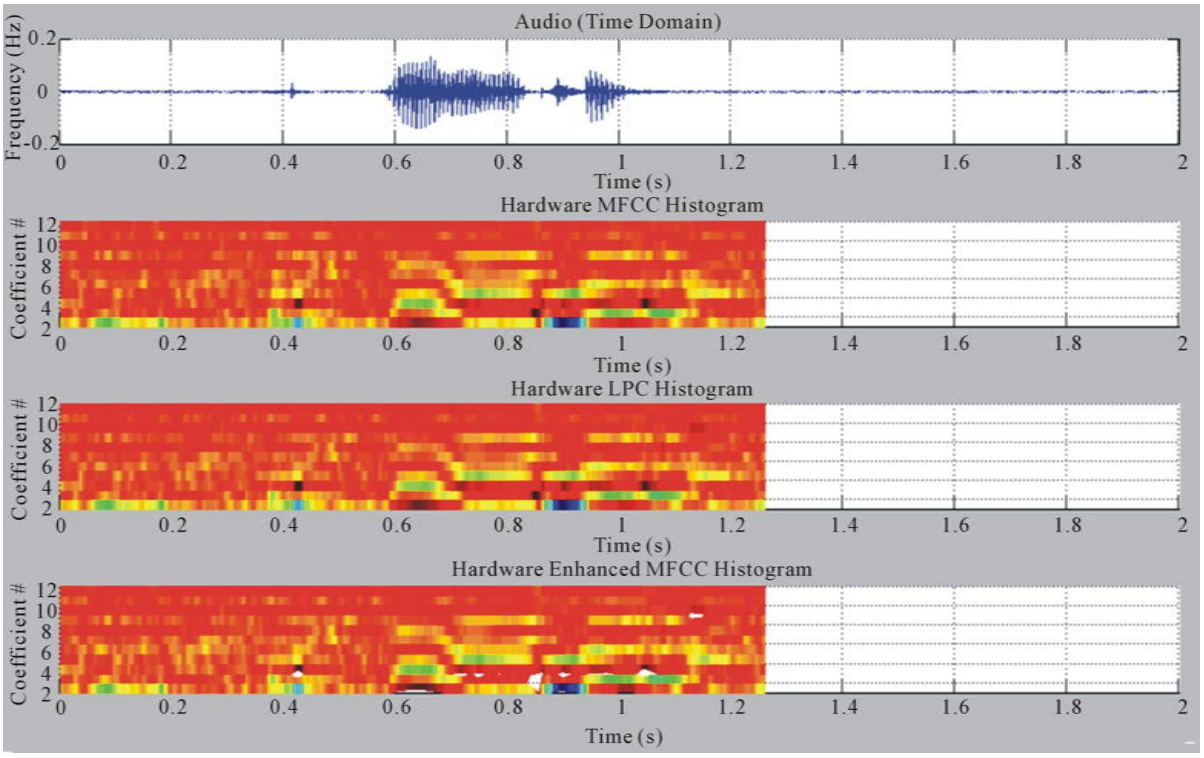

Figure 9. Hardware frontend histograms with audio data (12_Coefficients) (due to limited amount of hardware resources the part of the data is not show in the resulting spectrograms).

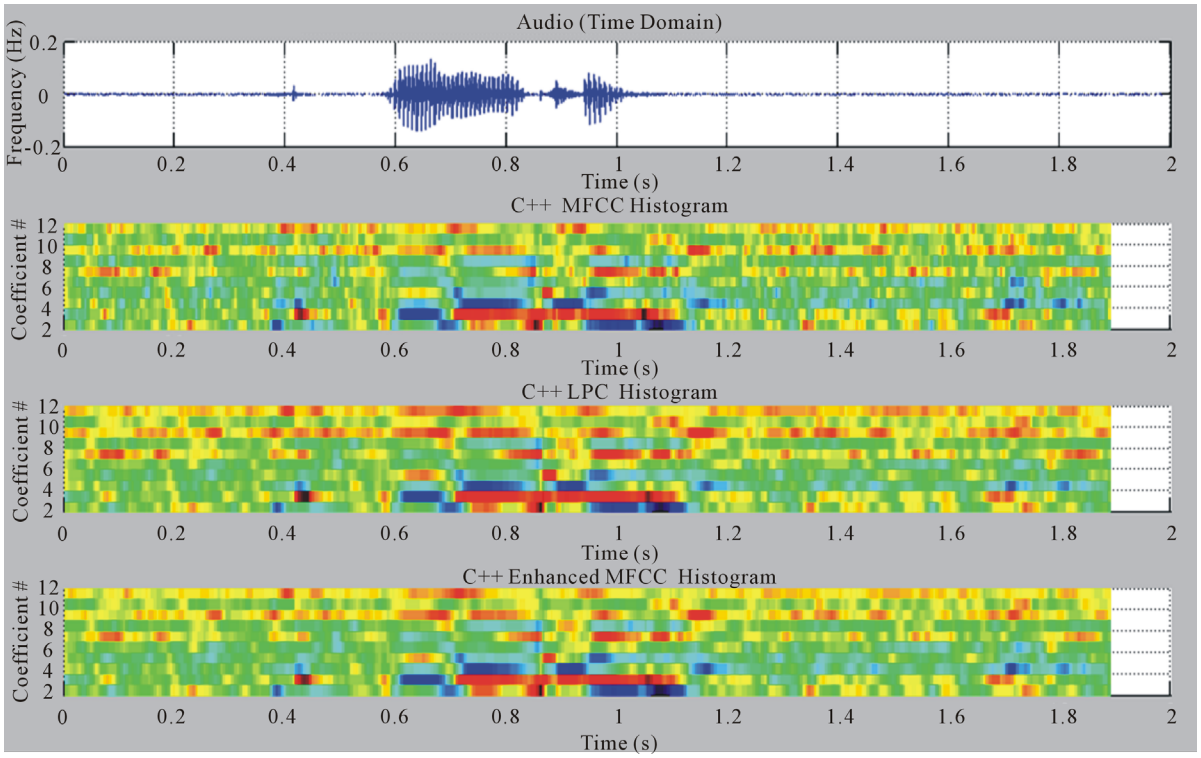

Figure 10. C++ frontend histograms with audio data (11_Coefficients C2-C12).

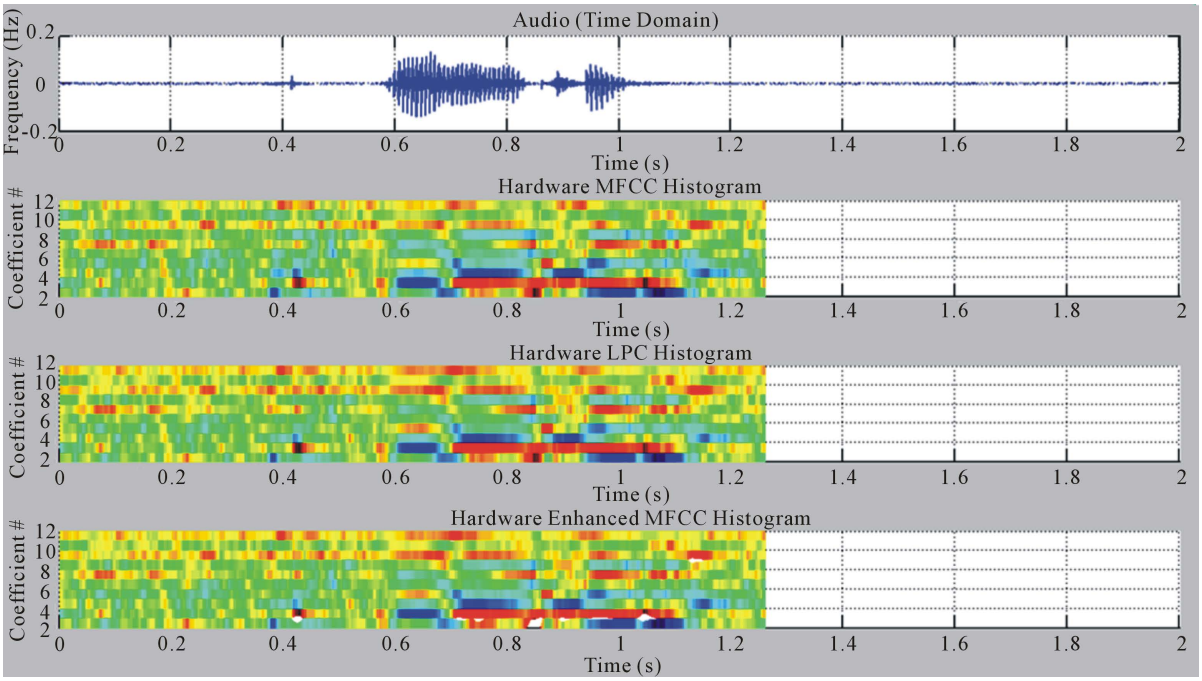

Figure 11. Hardware frontend histograms with audio data (11_Coefficients C2-C12) (due to limited amount of hardware resources the part of the data is not show in the resulting spectrograms).

ware and Software (C++) are identical.

9. Conclusions and Applications

In this study, the efficient hardware architecture and implementation of front-end of WUW-SR in FPGA is presented. This front-end is responsible for generating three sets of features MFCC, LPC, and ENH-MFCC. These features are needed to be decoded with corresponding HMMs in the back-end stage of the WUW Speech Recognizer (e.g., server). WUW Speech Recognition presented is a novel solution. The most important characteristic of a WUW-SR system is that it should guarantee virtually 100% correct rejection of non-WUW (out of vocabulary words—OOV) while maintaining correct acceptance rate of 99% or higher (in vocabulary words—INV). This requirement sets apart WUW-SR from other speech recognition systems because no existing system can guarantee 100% reliability by any measure. The computational complexity and memory requirement of three features algorithms are analyzed in detail in the past showing a significant improvement [2].

The partitioned table look-up method is adopted and modified to be suitable in our case with very small table memory requirement. The overall performance and area are highly improved. The proposed front-end is the first hardware system specifically designed for WUW-SR feature extraction based on three different sets of features. To demonstrate its effectiveness, the proposed design has been implemented in cyclone III FPGA hardware. The custom DSP board developed is a power-efficient, flexible design and can also be used as a general-purpose prototype board.

References

[1] Këpuska, V.Z., Eljhani, M.M. and Hight, B.H. (2013) Front-end of wake-up-word speech recognition system design on FPGA. *Journal of Telecommunications System & Management*, **2**, 108.

[2] Këpuska, V.Z. and Klein, T.B. (2009) A novel wake-up-word speech recognition system, wake-up-word recognition task, technology and evaluation. *Nonlinear Analysis, Theory, Methods & Applications*, **71**, e2772-e2789.

[3] Tuzun, O.B., Demirekler, M. and Bora, K. (1994) Comparison of parametric and non-parametric representations of speech for recognition. 7*th Mediterranean Electrotechnical Conference*, Antalya, 12-14 April 1994, 65-68.

[4] Openshaw, J.P., Sun, Z.P. and Mason, J.S. (1993) A comparison of composite features under degraded speech in speaker recognition. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, **2**, 371-374. http://dx.doi.org/10.1109/ICASSP.1993.319316

[5] Vergin, R., O'Shaughnessy, D. and Gupta, V. (1996) Compensated mel frequency cepstrum coefficients. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Minneapolis, 7-10 May 1996, 323-326.

[6] Davis, S. and Mermelstein, P. (1980) Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **28**, 357-366. http://dx.doi.org/10.1109/TASSP.1980.1163420

[7] Combrinck, H. and Botha, E. (1996) On the mel-scaled cepstrum. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.18.1382&rep=rep1&type=pdf

[8] Schroeder, M.R. (1982) Linear prediction, extremely entropy and prior information in speech signal analysis and synthesis. *Speech Communication*, **1**, 9-20. http://dx.doi.org/10.1016/0167-6393(82)90004-8

[9] Paliwal, K.K. and Kleijn, W.B. (1995) Speech synthesis and coding, chapter quantization of LPC parameters. Elsevier Science Publication, Amsterdam, 433-466.