ISSN: 2167-0919

Journal of Telecommunications System & Management

The International Open Access
Journal of Telecommunications System & Management

Available online at: OMICS Publishing Group (www.omicsonline.org)

Digital Object Identifier: http://dx.doi.org/10.4172/2167-0919.1000108

Telecommunications System & Management

Këpuska et al., J Telecommun Syst Manage 2012, 2:1
http://dx.doi.org/10.4172/2167-0919.1000108

Research Article Open Access

Front-end of Wake-Up-Word Speech Recognition System Design on FPGA

Veton Z Këpuska*, Mohamed M Eljhani and Brian H Hight

Electrical & Computer Engineering Department, Florida Institute of Technology, Melbourne, FL 32901, USA

Abstract

A typical speech recognition system is push button operated (Push-to-talk), which requires hand movement and hence mixed multi-modal interface. However, for disabled patients and those who use hands-busy applications (e.g., where the user has objects to manipulate or device to control while asking for assistance from another device) movement may be restricted or impossible. One alternative is to use Speech Only Interface. The method that is being proposed is called Wake-Up-Word Speech Recognition (WUW-SR). A WUW-SR system would allow the user to operate (activate) many systems (Cell phone, Computer, Elevator, etc.) with speech commands instead of hand movements. This paper introduces a new front-end paradigm of the Wake-Up-Word Speech Recognition. The state of the art WUW-SR system is based on three different sets of features: (1) Mel-frequency Cepstral Coefficients (MFCC), (2) Linear Predictive Coding Coefficients (LPC), and (3) Enhanced Mel-frequency Cepstral Coefficients (ENH_MFCC), these features are decoded with corresponding Hidden Markov Models (HMMs) in the back-end stage of the WUW-SR. We present an experimental FPGA design and implementation of a novel architecture of a real time feature extraction processor that generates MFCC, LPC, and ENH_MFCC features simultaneously. In the WUW-SR system, the recognizer front-end is located at the terminal which is typically connected over a data network to remote back-end recognition (e.g., server). The three sets of feature extraction of speech (MFCC, LPC, and ENH-MFCC) are performed at the front-end. These extracted features are then compressed and transmitted to the server via a dedicated channel, where subsequently they are decoded. Our front-end can be added to any hand-held electronic device compatible with WUW-SR and command (activate) 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 acting as a processor that is capable of computing three different sets of features at a much faster rate than software. It is cost effective, 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.

Keywords: Speech Recognition System; Mel-frequency Cepstral Coefficients; Linear Predictive Coding Coefficients; Enhanced Mel-frequency Cepstral Coefficients; Hidden Markov Models; Field-Programmable Gate Array

Introduction

"Operator (WUW for Elevator Simulator)! Take me to the last floor." "Operator" responds by "Taking you to the last floor." The ideas of being able to talk to a machine and have it understand you have been a reoccurring theme in science fiction for decades. While we are not yet at the stage where electronic machines can comprehend our every word and act as necessary on it, these machines are becoming ever more complex and ubiquitous.

For the past several decades, designers have processed speech for a wide variety of applications ranging from mobile communications to automatic reading machines. Speech has not been used much in the field of electronics and computers due to the complexity and verity of speech signals. However with modern processes and methods we can process speech signals in Field-Programmable Gate Array (FPGA) chips. While others concentrate on developing the algorithms and models, there still remains the question of how to implement them on portable device. Several speech recognition software packages already exist including the Wake-Up-Word Speech Recognition System that can run on a PC successfully; however, they are limited by having to operate on a general-purpose processor. In the end, to achieve the maximum processing power, application-specific hardware is the answer. A great deal of work has been conducted in this paper to address this problem by designing an efficient hardware front-end of state of the art WUW-SR [1] with an FPGA using an Altera DSP-based system, acting as a processor that is responsible for extracting three different sets of features from the input audio signal. These features are Mel-frequency Cepstral Coefficients (MFCC), Linear Predictive Coding Coefficients (LPC), and Enhanced Mel-frequency Cepstral Coefficients (ENH-MFCC).

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 [2]. In recent studies of speech recognition system, the MFCC parameters perform better than others in the recognition accuracy [3,4]. This paper presents the feature extraction solution based on LPC, MFCC and new set of features named Enhanced Mel-frequency Cepstral Coefficients (ENH –MFCC) with the architecture specially optimized for implementation in FPGA structures. The presented system is designed to be implemented in FPGA device as a System-on- Programmable-Chip (SOPC). This design not only has a relatively low resource usage, but also maintains a reasonably high level of performance. The remainder of this paper is organized as follows. Section II describes the Wake-Up-Word speech recognition architecture. Section III describes the front-end of WUW-SR design procedure and architecture. Section IV describes the Mel-Frequency Cepstrum Coefficients (MFCC) algorithm. Section V describes the

***Corresponding author:** Veton Z. Këpuska, Electrical and Computer Engineering Department, Florida Institute of Technology, Melbourne, FL 32901, USA, E-mail: vkepuska@fit.edu

Received June 11 2013; **Accepted** July 10, 2013; **Published** July 12, 2013

Citation: Këpuska VZ, Eljhani MM, Hight BH (2013) Front-end of Wake-Up-Word Speech Recognition System Design on FPGA. J Telecommun Syst Manage 2: 108. doi:10.4172/2167-0919.1000108

Autocorrelation Linear Predictive Coding (LPC) algorithm. Section VI describes the Enhanced Mel-Frequency Cepstrum Coefficients (ENH-MFCC) algorithm. In section VII the results and comparisons of three features spectrogram from MATLAB, and FPGA hardware implementation are described and compared with the C++ front-end algorithm. These are followed by conclusions in Section VIII.

WUW-SR System Architecture

As shown in Figure 1, the WUW-SR can be broken down into three components [1]. 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 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.

Front-end of WUW-SR System Architecture

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

Stage A: Pre-processing

a. Analog to Digital Converter ADC.

b. DC Filtering.

c. Serial to 32-bit parallel converter.

d. Integer to floating-point converter.

e. Pre-emphasis filtering.

f. Window advance buffering.

g. Hamming window.

Stage B: Linear predictive coding coefficients

The five brown-colored modules represent the Linear Predictive Coding Coefficients (LPC) stage and are used to generate 13-Linear Predictive Coding features.

a. Autocorrelation Linear Predictive Coding.

b. Fast Fourier Transform FFT.

c. LPC Spectrogram.

d. Mel-scale Filtering.

e. Discrete Cosine Transform DCT.

Stage C: Mel-frequency cepstral coefficients

The four pink-colored modules represent the MFCC stage and are used to generate 13 MFCCs features.

a. Fast Fourier Transform FFT.

b. MFCC Spectrogram.

c. Mel-scale Filtering.

d. Discrete Cosine Transform DCT.

Stage D: Enhanced Mel-frequency cepstral coefficients

The four green-colored modules represent the ENH-MFCC stage and are used to generate 13 ENH-MFCC features.

a. Enhanced Spectrum (ENH).

b. Enhanced MFCC Spectrogram.

c. Mel-scale Filtering.

d. Discrete Cosine Transform DCT.

Stage E: Voice activity detector

The five blue-colored modules represent the Voice Activity Detector (VAD) stage. The VAD is responsible for finding utterances spoken in the correct context and segmenting them from the rest of

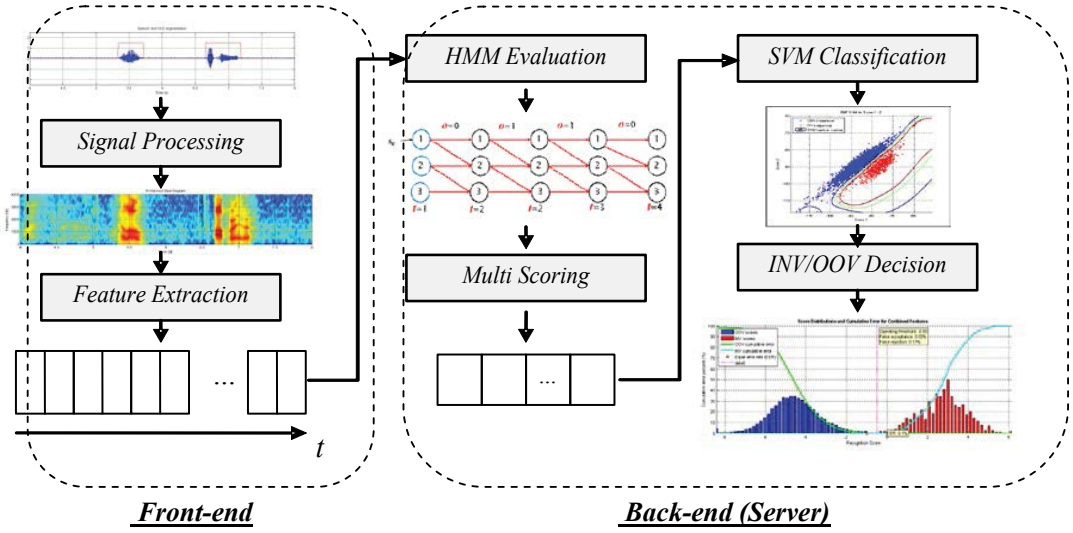

Figure 1: Overall WUW-SR architecture.

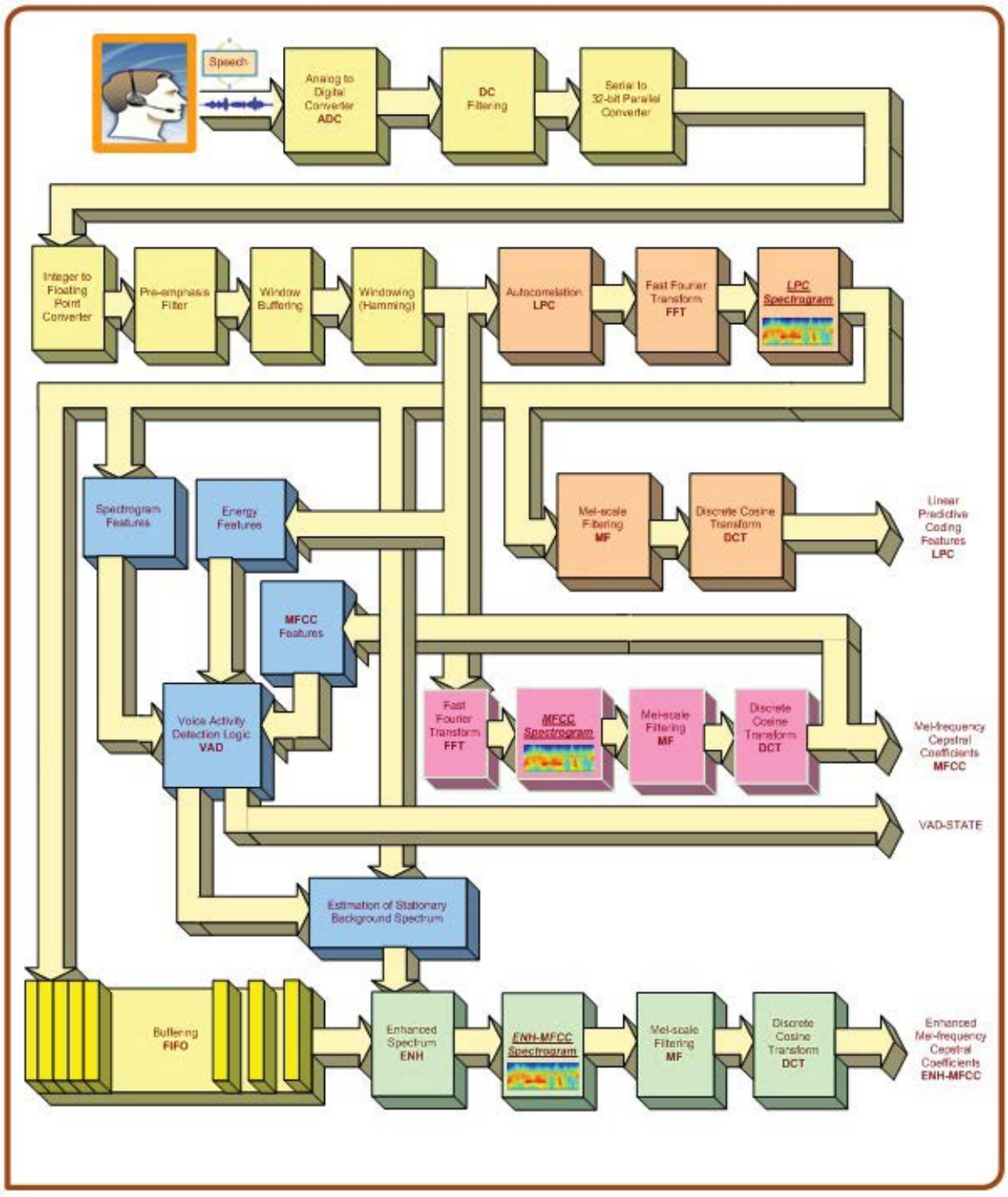

Figure 2: Front-end of WUW-SR Block Diagram.

the audio stream, then the system will identify whether or not the segmented utterance is a WUW.

a. Spectrogram features.

b. Energy features.

c. MFCC features.

d. Voice activity detection logic VAD.

e. Estimation of stationary background spectrum.

"FIFO module" is used to synchronize the LPC spectrum data with the Estimation Stationary Background Spectrum output data by 20 frames (160 msec).

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 [5]. 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 [6].

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.

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.

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.

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).

Serial to parallel converter

This model gets serial digital signal from CODEC and converts it to 32-bit.

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.

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:

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

*Output = Input – (PRE_EMPH_FACTOR * Previous_input)*

where we have chosen the value of *PRE_EMPH_FACTOR* ***(α)*** as 0.975.

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 (64 samples) to be multiplied by the Hamming coefficients, which are stored in a ROM memory.

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 N milliseconds, typically 32 ms wide (256 samples). A Hamming window size of 32 ms which consists of 256 samples at 8 KHz sampling frequency and 8 ms frame shift (64 samples) is picked for our front-end windowing.

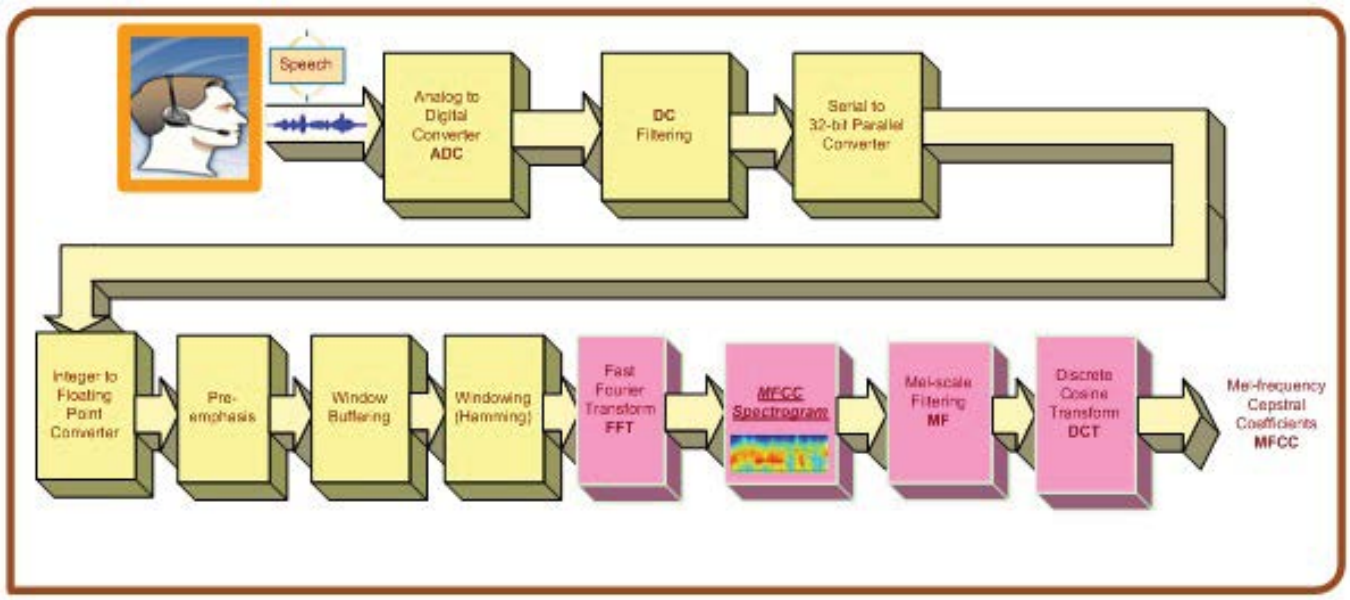

Figure 3: MFCC Feature Extraction.

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.

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.

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.

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.

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

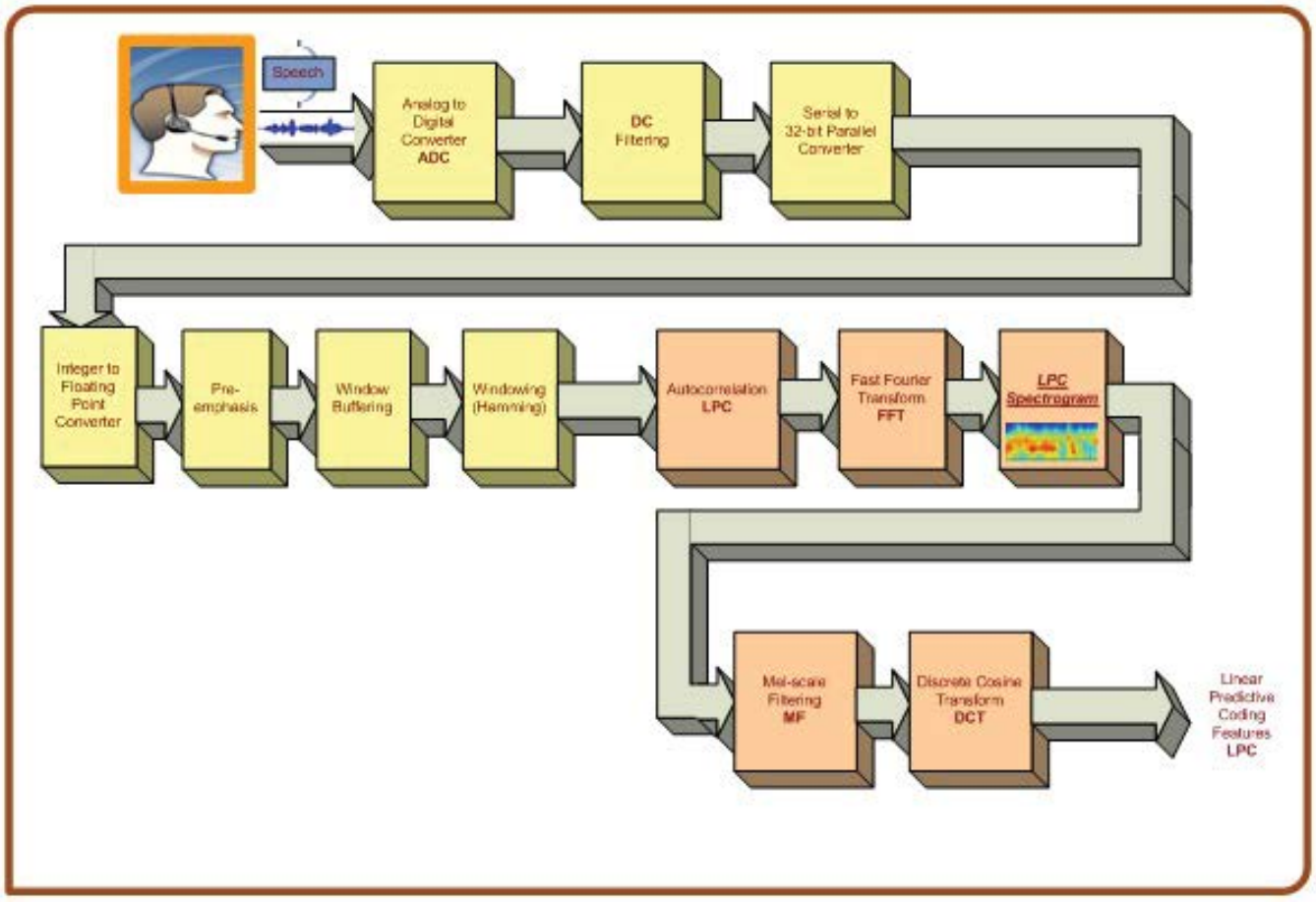

Figure 4: LPC Feature Extraction.

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 LPC algorithm is executed as follows:

```
for i = 1:17
% calculate the sum
for j = 1: (i-1)
s = s + lpc_ram(j) * acc_ram(i-j);
end
% calculate k
k = - (acc_ram(i) + s) / alpha;
% calculate lpc_ram[1:(i-1)]
for j = 1:(i-1)
temp_ram(j) = lpc_ram(j) + (k * lpc_ram(i-j)));
end
for j = 1:(i-1)
lpc_ram(j) = temp_ram(j);
end
% store new value of lpc_ram[i] and calculate new alpha
lpc_ram(i) = k;
alpha = alpha * (1-k*k);
% next iteration
end
```

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 [7]. 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 [8].

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 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 [1].

The algorithm uses only the single-sided spectrum, so the state machine starts the calculations when 128 data points have been written into the input RAM. The ENH-MFCC algorithm is executed as follows:

```
% "Silence Factor" Computation
normz=0;
for i=1:SPECL
normz=normz+temp_spec_vector(i);
end
normz=SF*normz+SIL_EN_FLOOR;
neighborhood_sum=zeros(SPECL,1);
% Computation of initial neighborhood sum for bin i=0
local_sum=temp_spec_vector(1);
for j=2:HALF_NEIGHB_SIZE+1
local_sum=local_sum+temp_spec_vector(j);
end
neighborhood_sum(1)=local_sum;
%Computing Neighborhood Sum X[j+i]
for i=2:SPECL
j=i+HALF_NEIGHB_SIZE;
k=i-HALF_NEIGHB_SIZE-1;
%Handling edge effects
if j >= SPECL
indx1 = SPECL;
else
indx1 = j;
end
if (k<1)
indx2 = 1;
else
indx2 = k;
end
%Adding New Element - Dropping Old one from local
local_sum=local_sum+temp_spec_vector (indx1)-temp_spec_vector(indx2);
%Removing Center Element from local_sum
neighborhood_sum (i)=local_sum-temp_spec_vector(i);
end
% Computing denominator
```

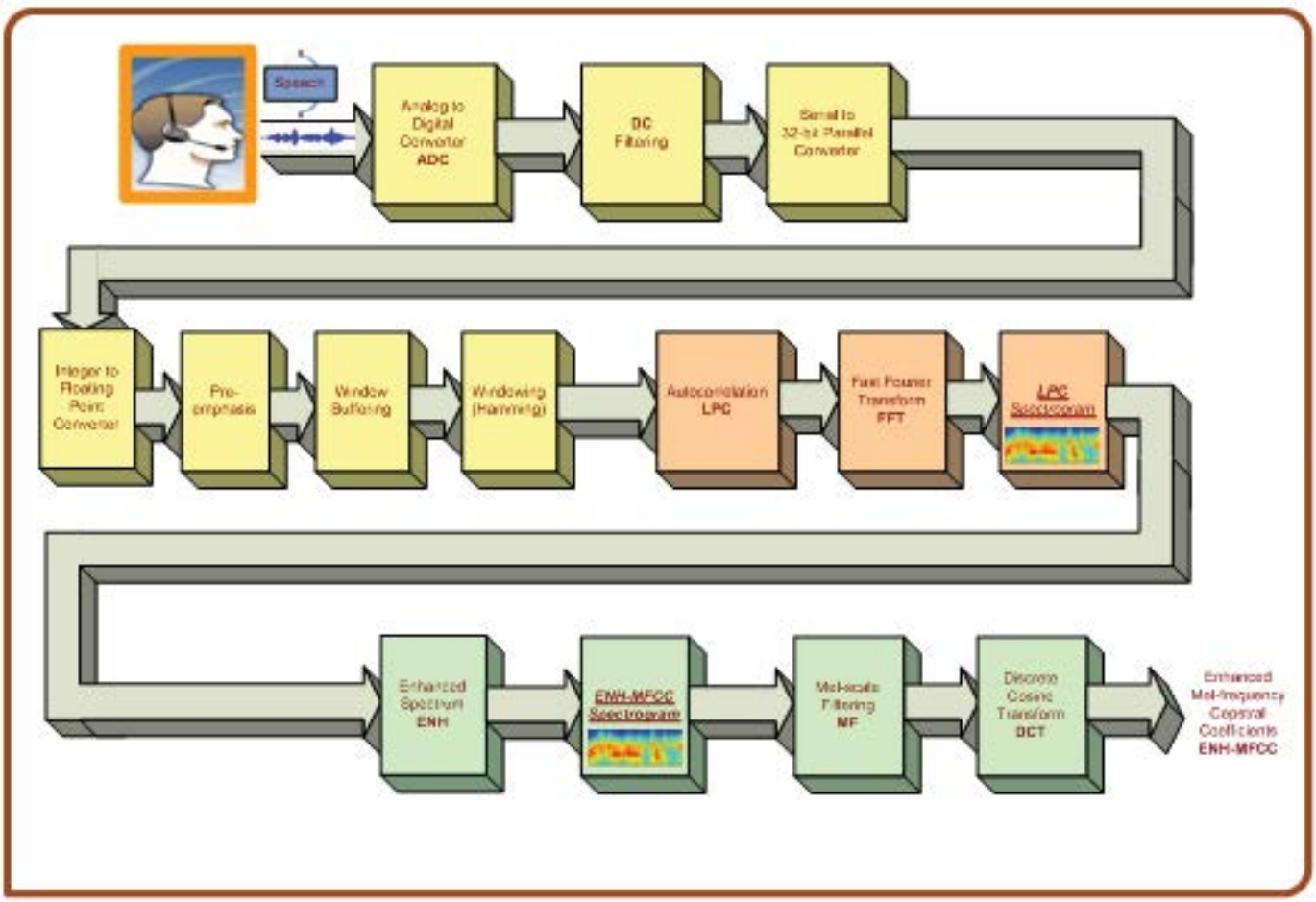

Figure 5: ENH-MFCC Feature Extraction.

```
denom=zeros(SPECL,1);
for i=1:SPECL
denom (i)=normz + NF*neighborhood_sum(i)+BG*background_estm(i);
end
% Scaling the output
out_spec_vector = zeros(SPECL,1);
for i=1:SPECL
tmp = EG * (temp_spec_vector(i) / denom(i));
out_spec_vector(i) = tmp * tmp;
end
```

The algorithm uses the following constants

```
LPC_INDEX_MAX= 8'h7F;                    // max. Spectrum index (128 total)
HALF_NBRHOOD_MAX= 8'h05;
SILENCE_FACTOR= 32'h3C23D70A;            // 1.0e-2
SIL_EN_FLOOR= 32'h501502F9;              // 1.0e10
NBRHOOD_FACTOR= 32'h3C23D70A;            // 1.0e-2
ENHANCE_GAIN= 32'h4CBEBC20;              // 1.0e8
```

Results and Comparisons

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 feature spectrograms (MFCC, LPC, and ENH-MFCC) with the software (C, C++) WUW's front-end algorithm implementation, and MATLAB front-end model which is implemented specially for this reason. 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 block by block based on software (C, C++) algorithm implementation and on equivalent floating-point MATLAB implementation. Each block was tested after it was completed to ensure correct operation before the next block was developed. The word "Onward" with 8KHz sampling rate was chosen as input audio data for testing our Front-end; we tested and compared (MFCC, LPC, and ENH-MFCC) spectrograms obtained from the hardware front-end model, with the MATLAB front-end model and, the software (C, C++) front-end model. The results show:

a. As shown in Figures 6-8, the MFCC, LPC, and ENH-MFCC

spectrograms generated from MATLAB, Hardware, and Software (C++) are identical and hence proper implementation.

b. In Figure 9, we generated the audio signal with the software (C++) front-end spectrograms that shows that they are sufficiently close with the Hardware front-end spectrograms (Figure 10).

Conclusions and Applications

In this study, we present an efficient hardware architecture and implementation of front-end of WUW-SR in FPGA. This front-end is responsible for generating three sets of features MFCC, LPC, and ENH-MFCC. These features are needed to be decoded with

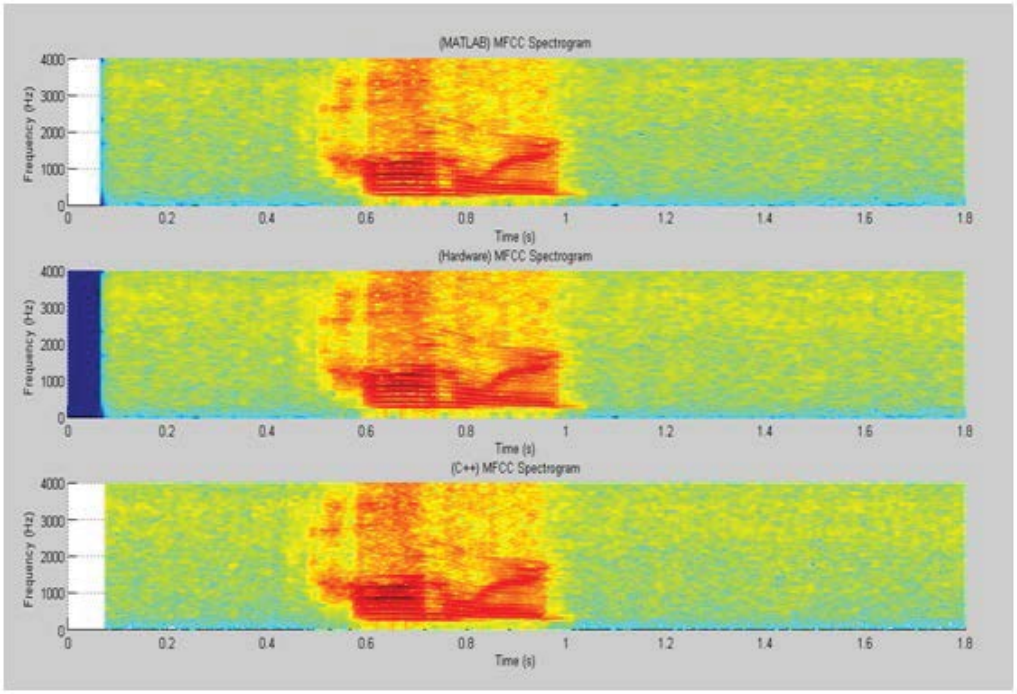

Figure 6: Front-end MFCC Spectrograms.

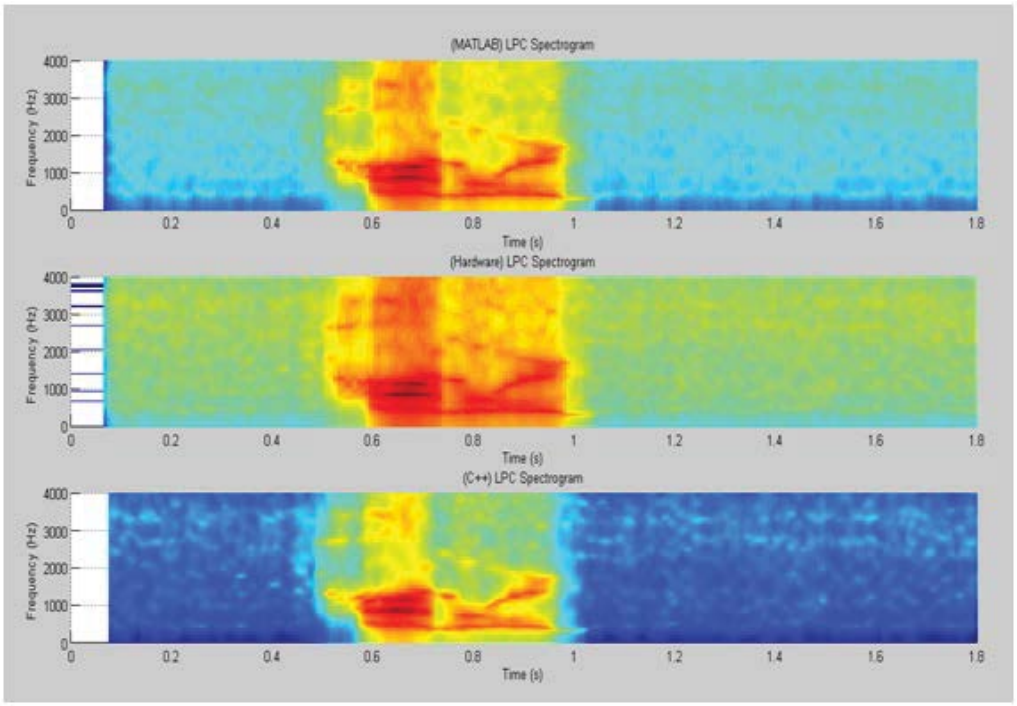

Figure 7: Front-end LPC Spectrograms.

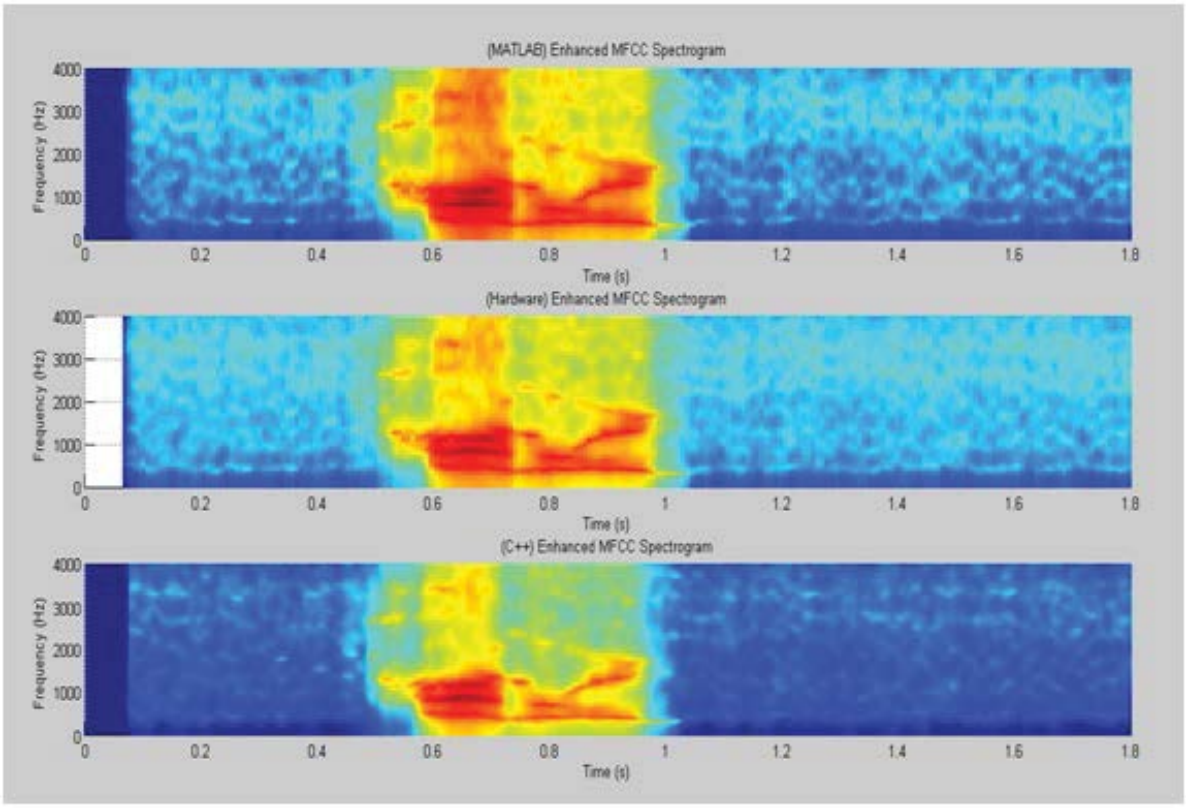

Figure 8: Front-end Enhanced MFCC Spectrograms.

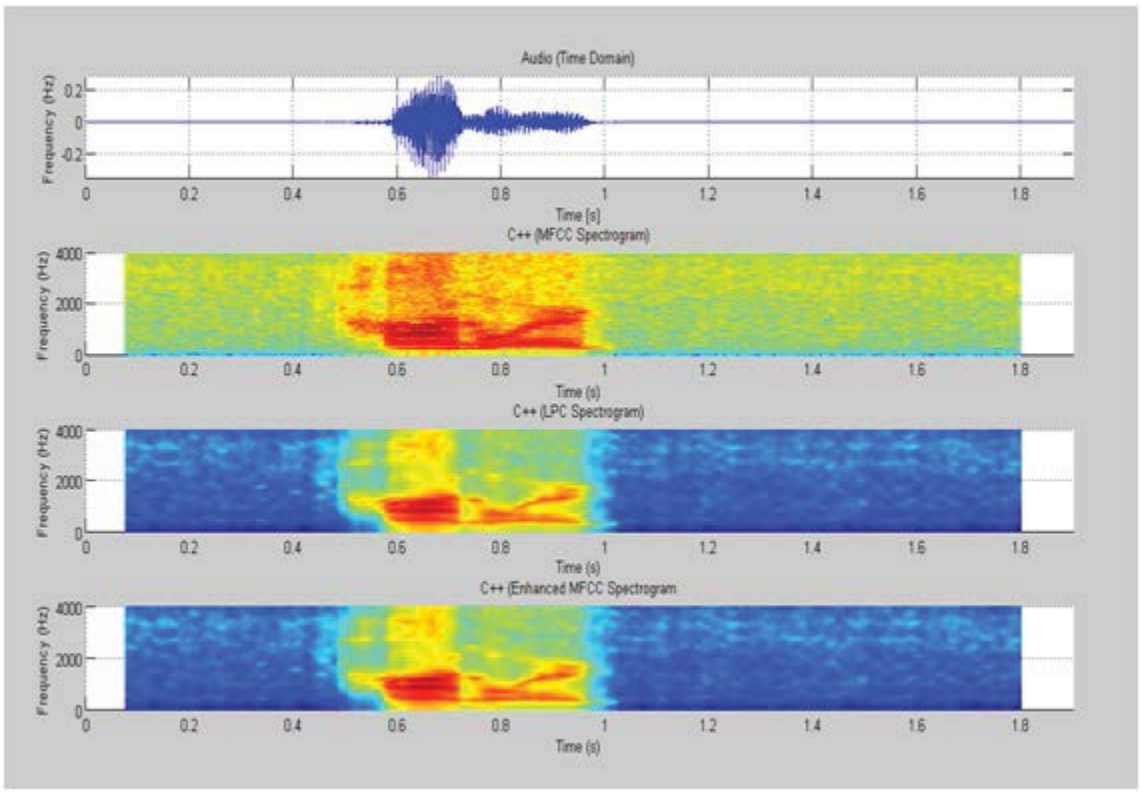

Figure 9: Software (C++) Front-end Spectrograms.

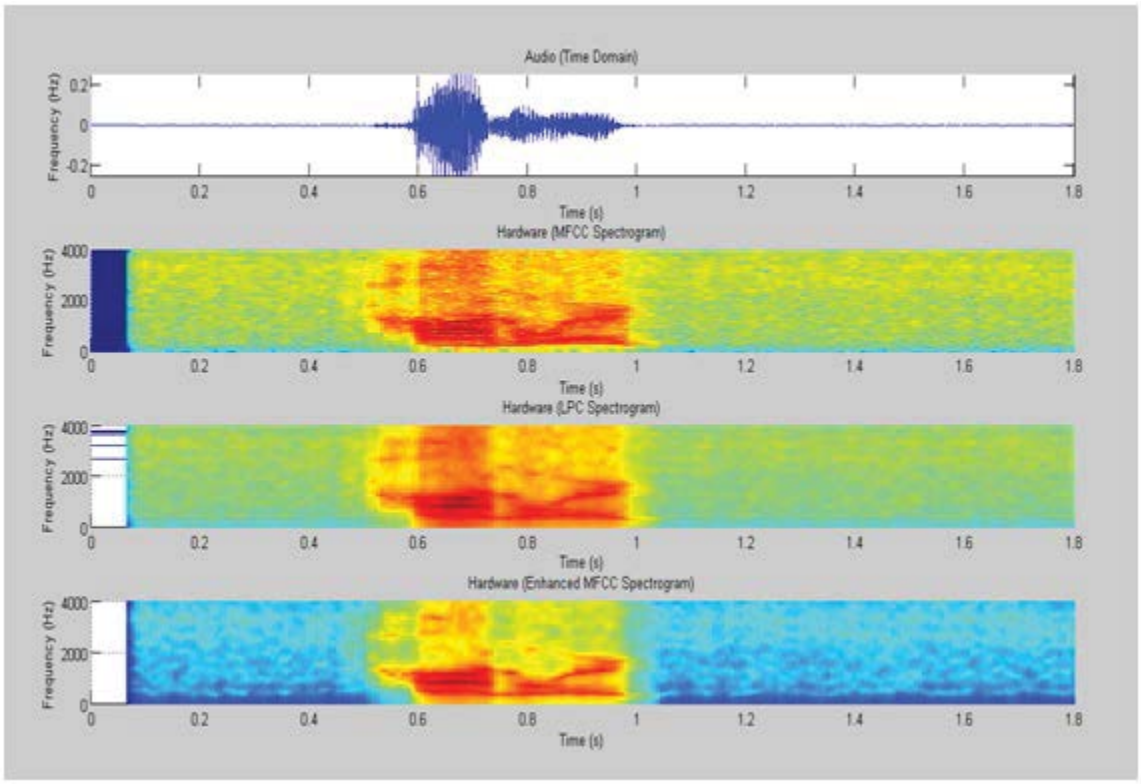

Figure 10: Hardware Front-end Spectrograms.

corresponding HMMs in the back-end stage of the WUW Speech Recognizer (e.g. server). WUW Speech Recognition presented 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 is analyzed in detail and improved greatly [1]. The partitioned table look-up method is adopted and modified to be suitable in our case with very small table memory. The overall performance and area is highly improved. The proposed front-end is the first hardware system specially designed for WUW-SR speech feature extraction based on three different sets of features algorithms. 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 VZ, Klein TB (2009) A novel Wake-Up-Word speech recognition system, Wake-Up-Word recognition task, technology and evaluation. Nonlin Anal Theory Meth Appl 71: e2772–e2789.
2. Burak O, Tuzun M, Demirekler M, Nakiboglu KB (1994) Comparison of Parametric and Non-Parametric Representations of Speech for Recognition. Proceedings of MELECON, Mediterranean Electrotechnical Conference 1: 65-68.
3. Openshaw JP, Sun ZP, Mason JS (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.
4. Vergin R, Shaughnessy DO, Gupta V (1996) Compensated mel frequency cepstrum coefficients. Proceedings of the International Conference on Acoustics, Speech, and Signal Processing 1: 323-326.
5. Davis S, Mermelstein P (1980) Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Trans Acoust Speech Signal Processing 28: 357-366.
6. Combrinck H, Botha E (1996) On the Mel-scaled Cepstrum. Department of Electrical and Electronic Engineering, University of Pretoria, South Africa.
7. Schroeder MR (1982) Linear prediction, extremely entropy and prior Information in speech signal analysis and synthesis. Speech Communication 1: 9-20.
8. Paliwal KK, Kleijn WB (1995) Quantization of LPC parameters. In Speech Synthesis and Coding. Elsevier Science Publ, Amsterdam, the Netherlands, 433-466.

Citation: Këpuska VZ, Eljhani MM, Hight BH (2013) Front-end of Wake-Up-Word Speech Recognition System Design on FPGA. J Telecommun Syst Manage 2: 108. doi:10.4172/2167-0919.1000108