Indian Journal of Science and Technology, *Vol 8(19), DOI: 10.17485/ijst/2015/v8i19/76703, August 2015*

*ISSN (Print) : 0974-6846*
*ISSN (Online) : 0974-5645*

# Real Time Implementation of Speaker Recognition System with MFCC and Neural Networks on FPGA

**Bhanuprathap Kari*** **and S. Muthulakshmi**

School of Electronics Engineering, VIT University,
Chennai-600127, Tamil Nadu, India;
bhanuprathap.kari2013@vit.ac.in, muthulakshmi.s@vit.ac.in

## Abstract

**Background**: Speaker recognition systems plays a pivotal role in the field of forensics, security and biometric authentication for verifying or identifying the speaker from the group of speakers. **Methods**: This paper gives a brief introduction about developing a hardware based speaker recognition system using Mel Frequency Cepstral Coefficients (MFCC) which are extracted from input speech signal to linearize the frequency scale at higher frequencies and Perceptron Neural Networks to provide layer weights for verifying the speaker identity to compare the output in the database of stored speaker identities. **Findings**: The input speech features are extracted using blocking and windowing to reduce noise and get the audio samples to store in the RAM where sampled data is converted into frequency domain using FFT to get the Cepstral Coefficients which are normalised and fed to neural network tool box present in the MATLAB to obtain layer weights for given set of data and the output is compared with the saved speaker identities to find a match. The decision making logic is written in NIOS II processor of FPGA where the taken input features are compared to the existing database of speaker identities with the help of perceptron neural network layer weights which gives the nearest possibility of the match in the database of the group of speakers. The designed system has been tested using two speakers as reference where the vowels spoken by them are taken into account to compare with the database of speakers already stored in FPGA. **Conclusion/ Improvements:** The probability of detection of the speakers is 80% and verifying the speaker is quite accurate in hardware based systems than in software based systems where performance factor is less. The given performance in the designed system can be increased by retraining the neural networks which can provide nearly 90% in detecting the speaker.

**Keywords:** Artificial Neural Networks, FPGA, MFCC, NIOS II, Speaker Recognition

## 1. Introduction

The speech characteristics of each and every human being varies because of their positional variation in voice box which consists of larynx, voice tract etc. As speech is one of the peculiar feature which is a part of human DNA, this interesting fact made a huge impact on the research field in this domain. To identify or to verify a particular speaker, the basic feature that is needed is training the acquired speech signals which is stored in database. So, when an unknown speaker speaks the used processor in the system has to compare the features present in the known database. The major features that is considered to obtain the speech characteristics is the fundamental frequency of an individual. By using this parameter the database can be obtained by training the cepstral coefficients. When human speech is analyzed in frequency domain the results can be used to identify the speaker. The taken input speech has oscillation of the vocal chords results in an underlying fundamental frequency and a series of harmonics. This value of fundamental frequency differs from speaker to speaker.

Speaker recognition comprises of different tasks like speaker diarization, speaker identification and speaker verification etc. In the process of speaker verification, the system must check whether the person is the one who

**Author for correspondence*

claims him to be. Similarly, in the process of speaker identification, the designed system must know the speaker whose information is already stored in the database. The designed prototype is only used to provide information on both speaker verification and identification.

Now-a-days the need of hardware based speaker recognition system are high because of its computational speed and accurate information on the speaker. This led to design an FPGA based hardware speaker recognition system which gives us chance to provide these smart device units with effective utilization of resources. Many speech recognition systems are there in today's market, but this designed system is more useful because of its computational speed and resource utilization. This paper provides an insight to the development of an Altera FPGA based system design as an effective hardware solution.

Capturing the speech on to FPGA is a complex phenomena where all the designed blocks are arranged as per interconnections one after the other and later speech coefficients has to be taken from NIOS II console window. The taken coefficients has to be trained with the neural networks which uses specific algorithm and those weights are to be taken into the code that is to be execute on NIOS II processor.

## 2. Literature Survey

The main features that are to be considered for obtaining the result on speech recognition systems is the fundamental frequency of the individual[1]. The other parameter that is needed to distinguish between the individuals is the vowel based fundamental frequency. The vowels that is spoken by any individual gives high magnitude in frequency domain than the consonants that is spoken at the same time[2]. This information will be used in the designed system to verify and identify the speakers. There are many algorithms and techniques for reducing the design parameters like verification time, identification time and computation time of the system without reduced performance[3]. Identification time and computational time depends upon the number of speakers and audio vectors spoken by the speaker. There are two types of models that are present in processing the speech. One is statistical modelling and dynamic modelling.

The method that is used to reduce the identification time is done using frame picking analysis[4] and can also be done using sliding window sampling for the purpose of identification. The speech that is taken after filtering and windowing the audio samples are considered to be audio vectors which can be used to point using vector quantization method[5] which finds out the Euclidean distance between the vectors and the decision is taken based on that characteristic which helps in attaining negligible error rate during the process of recognition[6]. Using a software based speaker recognition system is necessary to gain knowledge on the process to train neural networks using MATLAB which is discussed[7].

The spectral analysis of the vowel is taken into account to analyse the magnitude in time domain and frequency domain which is given in Figure 1 and Figure 2. Based on these parameters the speaker can be verified and identified in this design.

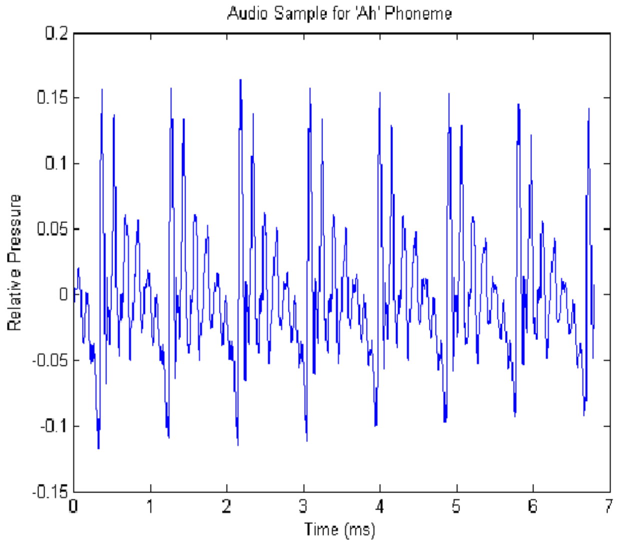


**Figure 1.** Audio sample for 'Ah' phoneme.

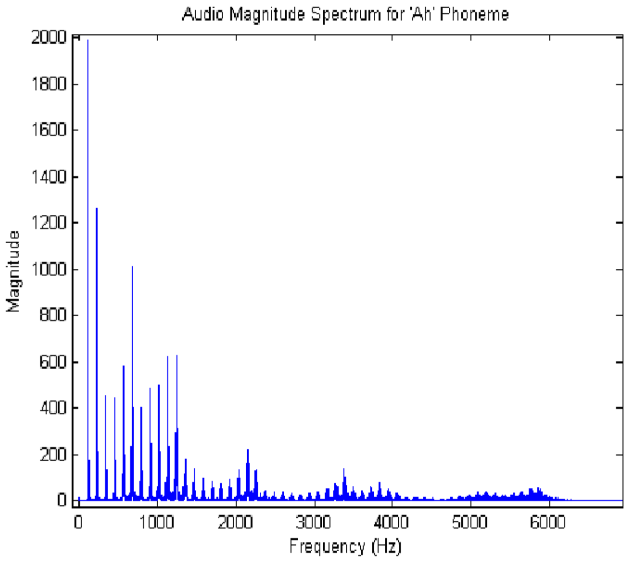


**Figure 2.** Magnitude rise for vowel sound.

By studying the sample speech signals it is known that vowels has high energy region which means the higher voltages, which helps to identify the speaker.

## 3. System Architecture

### 3.1 Block Diagram

Input audio data is taken through microphone of FPGA and stored in Audio RAM after applying filtering and windowing techniques which results in providing audio vectors. Later these audio vectors are converted into frequency domain for analyzing the magnitudes of the vowels spoken by particular person. The obtained coefficients are then converted into Mel domain to linearize the high frequencies. Normalize those frequencies and compare with the trained database to identifying the speaker which is depicted using pass/ fail indicators which are green led and red led respectively. Figure 3 shows the block diagram of the designed system which gives the information about the processing steps of taken audio samples.

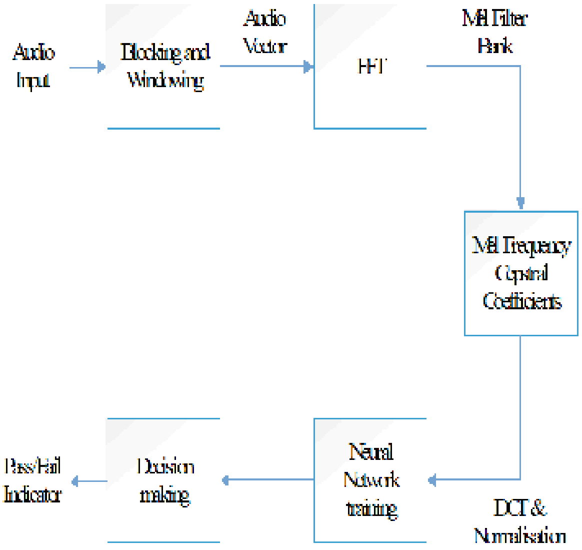


**Figure 3.** Block diagram of the designed system.

### 3.2 Hardware Components

### *3.2.1 Altera DE2-115 Cyclone IV E FPGA*

Altera DE2-115 board consists of features for high end applications like audio, image and video applications. Few features which are needed for speaker recognition system are given below

- Altera Serial JTAG Configuration device
- Two 64MB SDRAM
- Avalon bus to communicate with arbitrated peripherals
- NIOS-II soft-core Processor
- 50MHz oscillator for clocking
- Audio codec of24-bitCD-quality with line-in, line-out, and microphone-in jacks

### *3.2.2 Microphone*

- Compatible with Altera FPGA.
- Low output ripple and noise.
- It has a standard 3.5mm jack.
- Supports 24-bit audio codec line in, line out of FPGA.

### 3.3 Software Components

### *3.3.1 Quartus II Software*

Quartus II is a CAD ISE tool which is used to design programmable logic devices for Altera and its devices. It incorporates an implementation of VHDL and Verilog for hardware description. The SOPC builder tool in this software automatically generates an interconnection logic between the utilized peripherals. Qsys tool in Quartus II optimizes network architecture.

### *3.3.2 Nios II Design Suite*

NIOS II Design Suite consists of customizable instruction set of NIOS II processor in it to execute the programs written in it. NIOS II processor is like a co-processor which is present only in Altera devices[8].It utilizes Avalon bus as an interface to communicate with its peripherals with the help of Qsys or SOPC builder tool. After selecting the program to run on hardware the Quartus II software will compile the entire system on to FPGA.

### *3.3.3 MATLAB*

The purpose of MATLAB in this system design is to train the neural networks using nftool and nntool[9].

## 4. System Implementation

### 4.1 Methodology

The speaker recognition task comprises of two steps

- Characteristic extraction
- Neural Network Training

### *4.1.1 Characteristic Extraction*

#### *4.1.1.1 Streamed Audio Data Input*

The initialization of Audio codec and its ADC and DAC unit to convert the actual speech signal into digital samples which works at 48KHz on the Altera DE2-115 Cyclone IV E FPGA kit. The Figure 4 shows the RTL Schematic gives the information of initialising the audio components required to drive Audio RAM and store the samples in SDRAM.

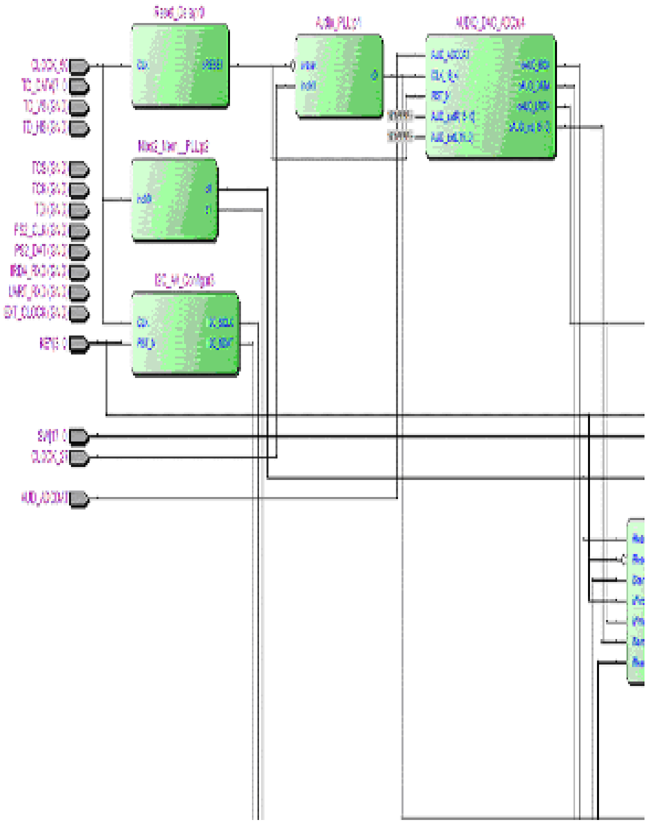


**Figure 4.** Initialising audio configuration.

## 4.2 Filtering

In this step the stored audio samples in the RAM has to be filtered using windows like hamming and hanning windows because of their applications in speech processing to reduce noise. Here the average of both the windows is taken to avoid background noise and performance ratio. These windowing techniques are used in the Audio RAM block of schematic which is shown in Figure 5.

## 4.3 FFT

This FFT helps the time domain signal to convert to frequency domain with less number of computations than DFT. The FFT gives same result as DFT with less number of computations as DFT computation is slower than FFT. FFT is implemented on FPGA using mega core functions present in Quartus II software. The equation for FFT is given as follows

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi k\frac{n}{N}} \quad k=0, 1,\ldots,N-1$$

Mega core functions of Quartus II are used to create FFT controller to synchronize with the voice recognizer block. Figure 6 gives the RTL schematic of FFT Controller.

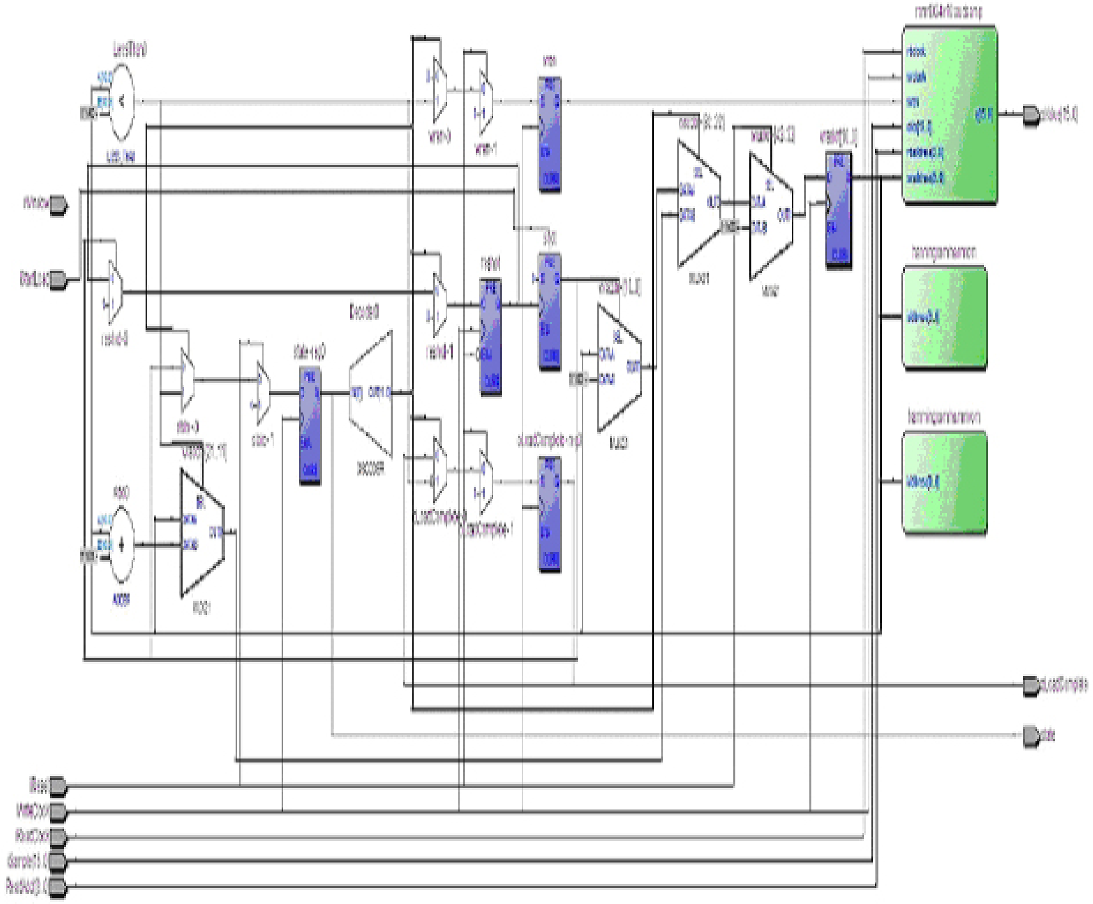


**Figure 5.** Part of Audio RAM with window ROM's.

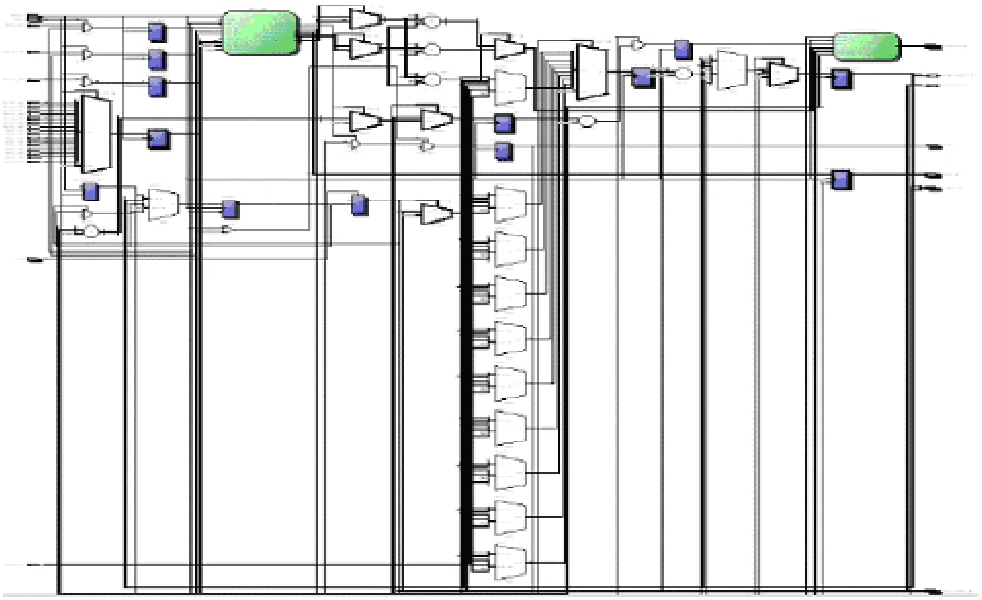


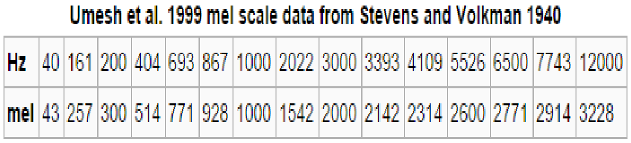
**Figure 6.** RTL schematic of FFT controller.

## 4.4 Mel scale

Mel domain is used to linearize the high frequencies to the normal frequencies with linear variation. By using the transfer function as given below, the higher frequencies are linearized.

$$m = 2595\log_{10}\left(1+\frac{f}{700}\right)$$

The Figure 7 gives information about the normal frequencies and Mel domain frequencies.



Umesh et al. 1999 mel scale data from Stevens and Volkman 1940

| Hz | 40 | 161 | 200 | 404 | 693 | 867 | 1000 | 2022 | 3000 | 3393 | 4109 | 5526 | 6500 | 7743 | 12000 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| mel | 43 | 257 | 300 | 514 | 771 | 928 | 1000 | 1542 | 2000 | 2142 | 2314 | 2600 | 2771 | 2914 | 3228 |

**Figure 7.** Mel scale.

### *4.4.1 Neural Network Training*

To train the MFCC, artificial neural network technique is used. MATLAB is essential in the design for the usage of neural network tool box present in it to train the networks for the weights of the neurons which helps in comparing the unknown speaker characteristics to the trained speaker characteristics. For this purpose nftool and nntool are used to train the data and to manage the data respectively. NNTOOL provides the user to manage the data for inputs, targets (which are nothing but our outputs), type of network used like back propagation algorithm, perceptron network, NARX, NAR and so on.

The neural network used here is perceptron network which is also called as feed forward neural network. The considered network consists of only one hidden layer and has five hidden neurons. The performance characteristics of the trained neural network is given Figure 8.

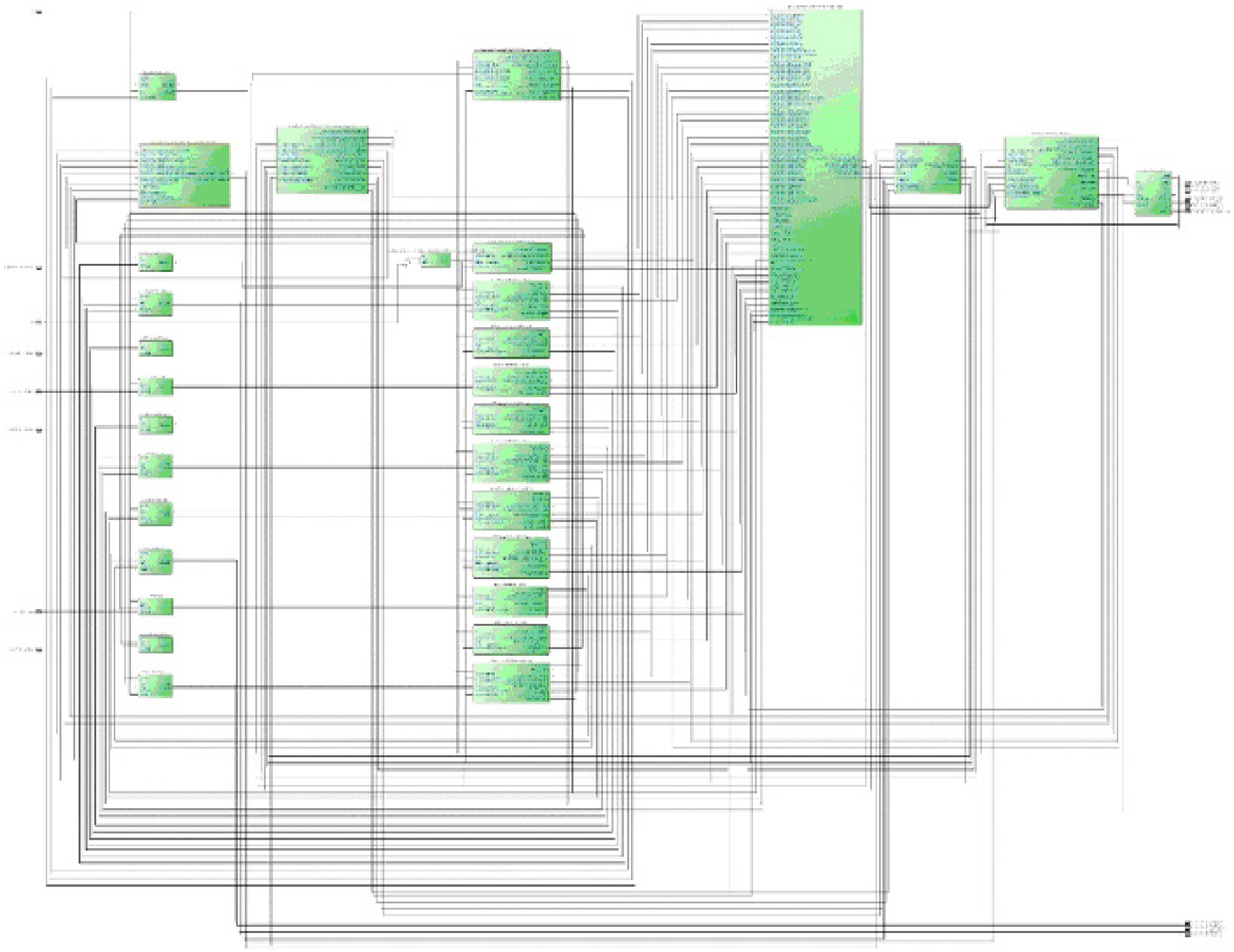


**Figure 8.** Voice recognizer block.

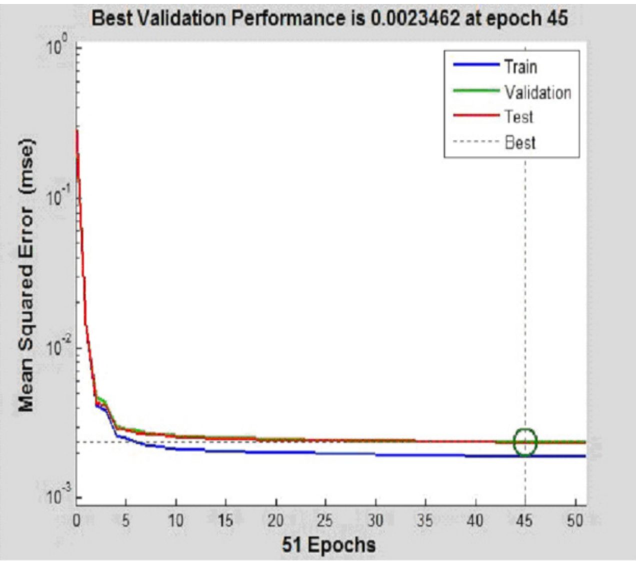


**Figure 9.** Performance charecteristics of trained neural network.

For any perceptron neural network, the output is given as follows y= tanh (b+ΣW$_i$*X$_i$). Where W$_i$ is weight of particular node and X$_i$ are the normalized inputs taken after performing DCT.

The performance plot of the trained neural network is given in the Figure 9.

The Regression plot of trained neural network after taking audio sample vectors from Nios II Console is plotted as shown in Figure 10.

### 4.5 Real Time Constraints

Problem faced during the interface of SDRAM component, as the clock did not synchronise with the SDRAM, the addresses accessed on the RAM did not match. So Altera Phase Locked Loop mega function in Quartus II is used to synchronise the clock.

During the FFT sampling phase, initially got infinite zeros as samples. This is because of asynchronization of values between FFT module and SDRAM module, sorted out this problem by using a two port RAM Mega function in Quartus II. Usage of this component also helped to store the float values of FFT and MFCC coefficient values in the memory.

Limited the number of samples to 12 for FFT and MFCC. This may have lowered the accuracy of the project to some extent and also the noise content into the microphone lead to some ambiguous results at one time. Hence increased number of training samples to 650 for each speaker.

## 5. Hardware Setup

The experimental setup is done using Altera DE2-115 Cylcone IV E FPGA board which is used for high end applications like image, audio and Ethernet etc. Altera devices consists of a 32 bit soft core processor named NIOS II which can act as a co-processor for parallel execution of the design.

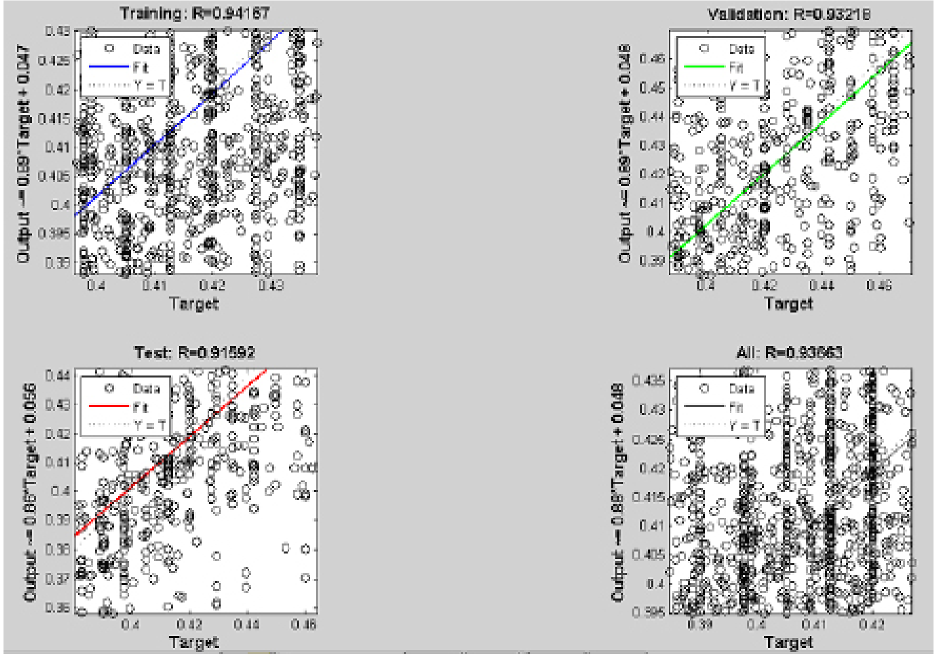


**Figure 10.** Performance plot using nftool.

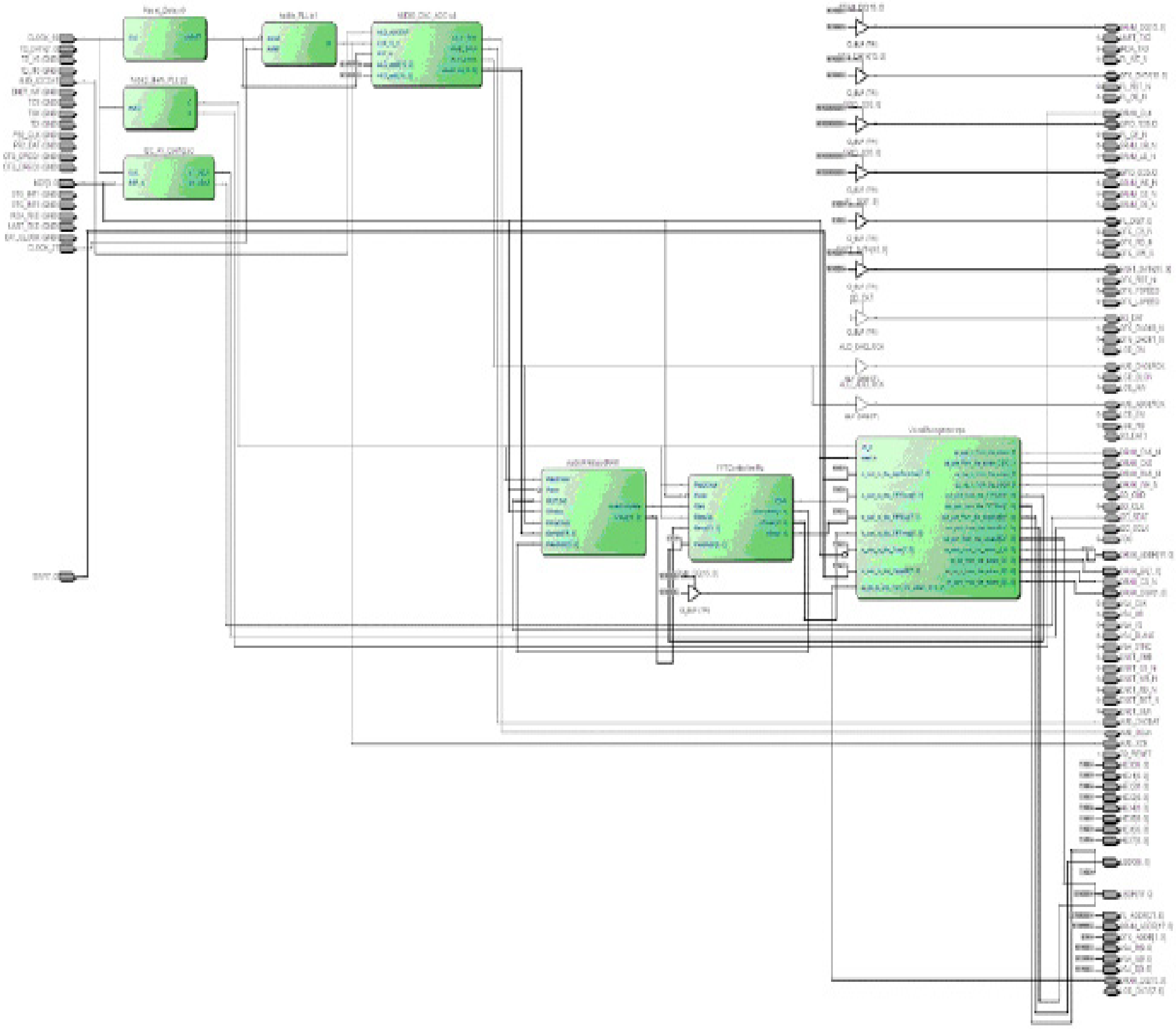
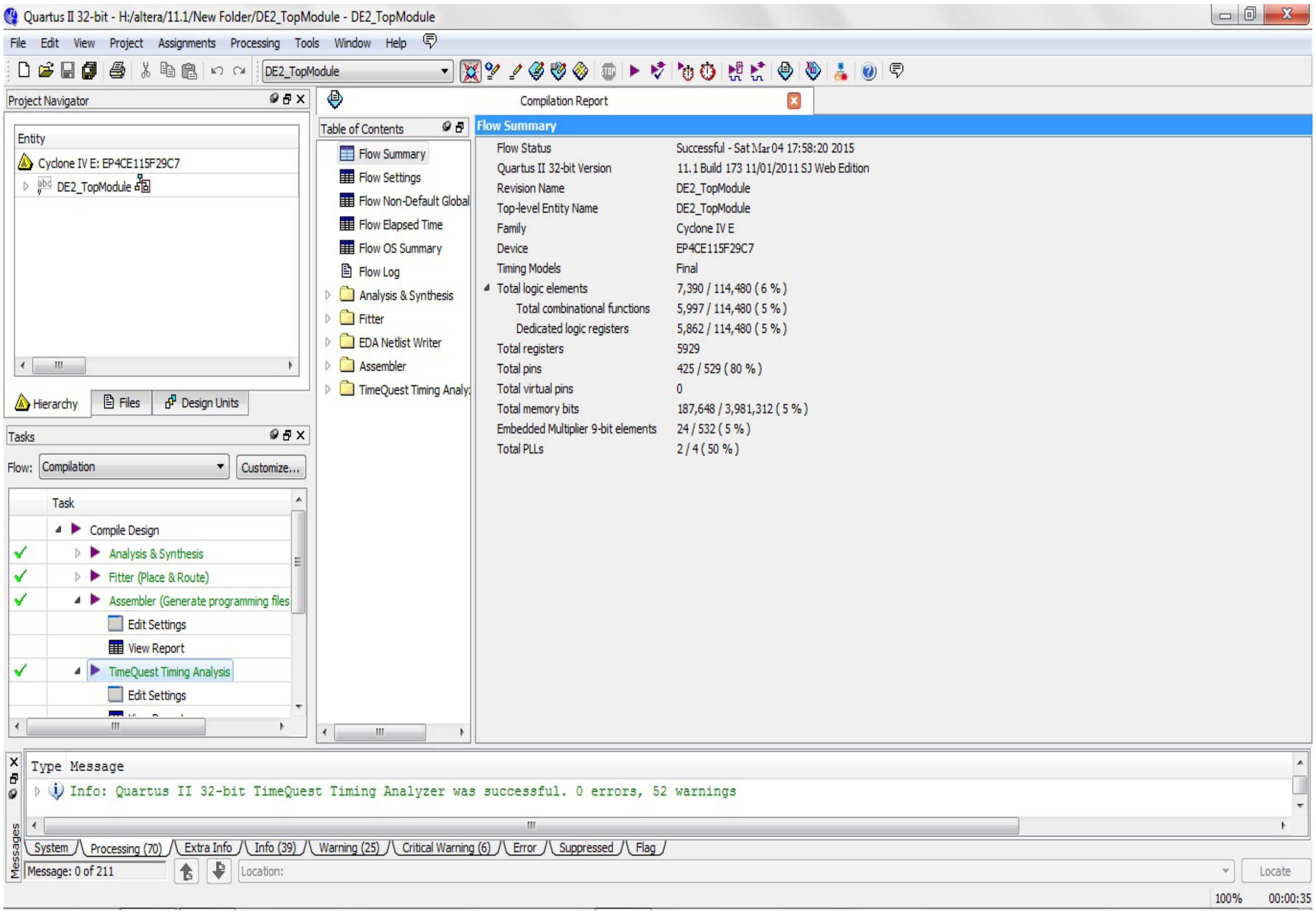


**Figure 11.** Top Module of the system.

In the design flow first two blocks, taking and storing input audio data from microphone and performing FFT to the filtered audio vectors are done on the FPGA. The remaining blocks like linearizing cepstral coefficients, normalizing cepstral coefficients and decision making are done using NIOS II soft core processor.

First the audio input data has to be taken from FPGA compatible microphone which is connected to the 24-bit audio codec mic of Altera DE2-115 board. These taken audio samples are stored in SDRAM of FPGA. After storing the samples, they have to be windowed using hamming and hanning windows to filter out the input coefficients. These obtained time samples are to be converted into frequency analysis for obtaining the actual cepstral coefficients using FFT. After the completion of FFT the coefficients are stored in SDRAM. Communication between NIOS II processor and FPGA is done using parallel I/O ports. To start the FFT process the NIOS II processor will give a signal called FFT start, and then the FFT operation for input windowed signal is started by FPGA. After completing the FFT the FPGA will give FFT done signal to the NIOS II processor. Then, a way is needed to read the results from RAM. By using PIO bus the addresses of the RAM are given and the data from that location are taken through the same PIO input bus.

When receiving the FFT done signal through the parallel I/O port to the voice recognizer block which consists



Flow Summary

| | |
|---|---|
| Flow Status | Successful - Sat Mar 04 17:58:20 2015 |
| Quartus II 32-bit Version | 11.1 Build 173 11/01/2011 SJ Web Edition |
| Revision Name | DE2_TopModule |
| Top-level Entity Name | DE2_TopModule |
| Family | Cyclone IV E |
| Device | EP4CE115F29C7 |
| Timing Models | Final |
| Total logic elements | 7,390 / 114,480 ( 6 % ) |
| Total combinational functions | 5,997 / 114,480 ( 5 % ) |
| Dedicated logic registers | 5,862 / 114,480 ( 5 % ) |
| Total registers | 5929 |
| Total pins | 425 / 529 ( 80 % ) |
| Total virtual pins | 0 |
| Total memory bits | 187,648 / 3,981,312 ( 5 % ) |
| Embedded Multiplier 9-bit elements | 24 / 532 ( 5 % ) |
| Total PLLs | 2 / 4 ( 50 % ) |

```
Type Message
Info: Quartus II 32-bit TimeQuest Timing Analyzer was successful. 0 errors, 52 warnings
```

**Figure 12.** Compilation report.

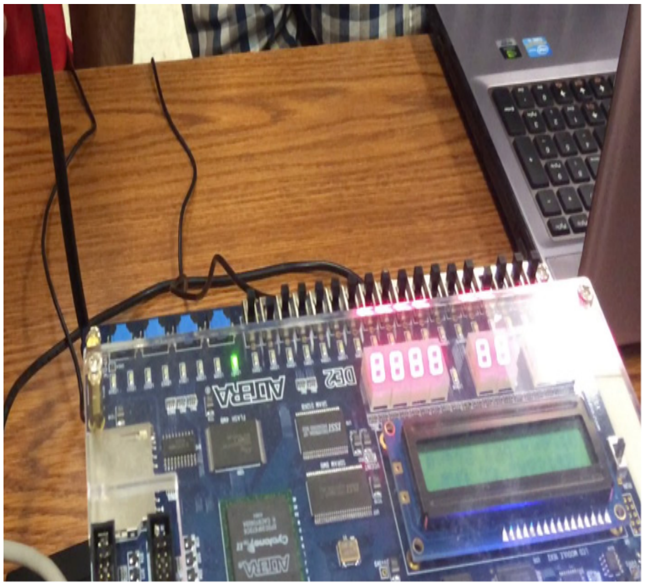


**Figure 13.** Speaker identified as speaker 1.

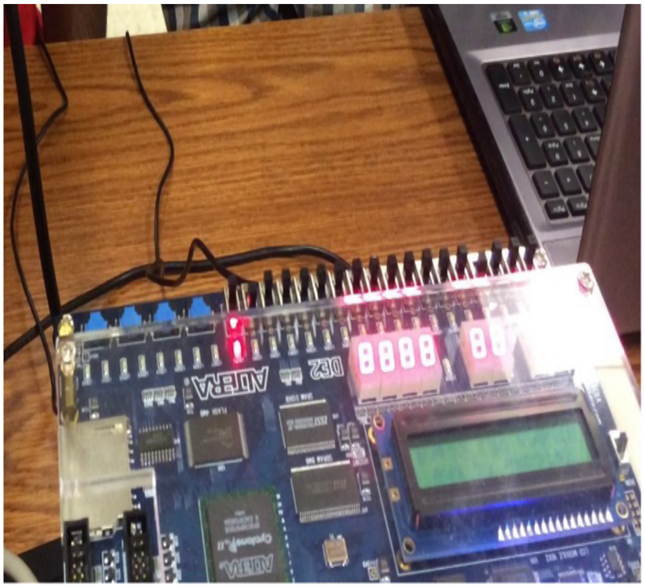


**Figure 14.** Speaker identified as speaker 2.

**Table 1.** Hit ratio of vowels of the designed system

| Speaker | Speaker 1 | | Speaker 2 | |
|---|---|---|---|---|
| Sounds | a | vowels | a | Vowels |
| Iterations | 30 | 1 each | 40 | 1 each |
| Detections | 30 | 'u' ceases detecting in 2nd-8th attempt | 30 | 4 ('i' not detected for 6 attempts) |
| Probability of Detection | 100% | 80% | 75% | 80% |

of NIOS II processor in it, processor will convert normal FFT frequencies to Mel domain. The obtained frequencies are normalized in between 0 and 1 and must be compared with the trained database.

In the process of decision making, the obtained normalized cepstral coefficients are compared to the known coefficients present in database to decide whether the spoken sound is vowel or consonant and to verify the person who he claims to be. To show the result to the user green and red led are used to verify and identify the speaker. If the green led is on then the speaker is authenticated or else if red led is on then the speaker is now not known and not from the trained database.

## 6. Results

The entire system implementation in RTL viewer is as shown in Figure 11. The designed system is tested for two speakers. When the first speaker uttered vowels with vowel ID 1 then it is recognizing by switching on the green led and when the second speaker uttered the same vowels with the vowel ID 2 then the designed system detected and shown the similar output by switching on green led.

Figure 12 provides the compilation report of the system. Figure 13 and Figure 14 gives information about acceptance and rejecting the speaker. The results are provided in a Table 1.

Coming to speaker identification when two speakers have uttered same vowels it detected by providing a name on LCD.

The compilation report of the entire system is given below which gives the information of utilization of available resources.

## 7. Conclusion

This paper gives a brief description to design a speaker recognition system on Alter DE2-115 FPGA. The results are obtained by training the neural networks and extracting the Mel cepstral coefficients form the speech signal taken through microphone. The designed system provides a performance of 80% which is satisfactory. This can be improved by retraining the neural networks.

The future scope of this work is to use statistical models like Gaussian Mixture Models and Hidden Markov Models to distinguish male speech from female speech, a process called speech diarization and some model based speech recognition tasks like speech segmentation and speech clustering.

## 8. References

1. AmrutaAM, Sahare S. Advanced speaker recognition. IJAET. 2012 July; 4(1):443–55. ISSN: 2231-1963
2. Saxena A, Sinha AK, Chakrawarti S, Surabhi C. Speech recognition using MATLAB. International Journal of Advances in Computer Science and Cloud Computing. 2013 Nov: 1(2):26–30. ISSN: 2321-4058
3. Selvan K, Joseph A, Babu KKA. Speaker recognition for security applications. 2013 IEEE Recent Advances, Intelligence Computational Systems; 2013 Dec 19-21; Trivandrum; p. 26–30.
4. Sarkar G, Saha G. Real Time Implementation of Speaker Identification System with Frame Picking Algorithm. Procedia Computer Science. 2010; 2: 173–80.
5. Priyanka M, Agrawal S. Recognotion of speaker useing mel frequency cepstral coefficient and vector quantization

for authentication. International Journal of Scientific and Engineering, Research. 2012 Aug; 3(8):1–6.

6. Patel K, Prasad RK. Speech recognition and verification using MFCC and VQ. International Journal of Emerging Science and Engineering. 2013 May; 1(7):33–7. ISSN: 2319–6378
7. Geeta N., Soni MK. Speaker recognition using support vector machine. International Journal of Computational Applications. 2014 Feb; 87(2):7–10.
8. Available from: www.altera.com/products/fpga/cyclone-series/cyclone-iv/support.html
9. Available from: http://in.mathworks.com/help/nnet/index.html