Journal of information and communication convergence engineering
http://jicce.org

J. Inf. Commun. Converg. Eng. 13(3): 145-151, Sep. 2015 **Regular paper**

FPGA-Based Hardware Accelerator for Feature Extraction in Automatic Speech Recognition

Chang Choo[1], Young-Uk Chang[1], and Il-Young Moon[2*], *Member, KIICE*

[1]Department of Electrical Engineering, San Jose State University, San Jose, CA 95192, USA
[2]School of Computer Science & Engineering, Korea University of Technology, Cheonan 31253, Korea

Abstract

We describe in this paper a hardware-based improvement scheme of a real-time automatic speech recognition (ASR) system with respect to speed by designing a parallel feature extraction algorithm on a Field-Programmable Gate Array (FPGA). A computationally intensive block in the algorithm is identified implemented in hardware logic on the FPGA. One such block is mel-frequency cepstrum coefficient (MFCC) algorithm used for feature extraction process. We demonstrate that the FPGA platform may perform efficient feature extraction computation in the speech recognition system as compared to the general-purpose CPU including the ARM processor. The Xilinx Zynq-7000 System on Chip (SoC) platform is used for the MFCC implementation. From this implementation described in this paper, we confirmed that the FPGA platform is approximately 500× faster than a sequential CPU implementation and 60× faster than a sequential ARM implementation. We thus verified that a parallelized and optimized MFCC architecture on the FPGA platform may significantly improve the execution time of an ASR system, compared to the CPU and ARM platforms.

Index Terms: ARM, Feature extraction, FPGA, MFCC, Automatic speech recognition, Zynq

I. INTRODUCTION

In recent times, the demand for speech recognition technology has dramatically increased for easier use of machines. The development of new interfaces performing exactly what humans want to do has been in the spotlight. Scientists first began to explore the possibilities of speech recognition in the 1970s, but because of the algorithm complexity, the development of speech recognition slowed down considerably. Then, in the late 2000s, the development of speech recognition picked up pace because of the use of high-speed computers, improvement in the digital signal process, and a drop in the prices of mass memories.

The mel-frequency cepstrum coefficient (MFCC) method has been widely used for feature extraction in automatic speech recognition (ASR). In the past few decades, the MFCC process was optimized for CPU-based ASR systems [1-5]. Recently, highly optimized MFCC algorithms for the Field-Programmable Gate Array (FPGA), Graphics Processing Unit (GPU), and Advanced RISC Machine (ARM) have been proposed. In particular, highly parallelized MFCC architectures on the FPGA and GPU platforms have been shown to exhibit very low execution times [6-12].

In [6], the MFCC process was implemented on the NVIDIA GTX580 GPU platform, which demonstrates a 90× speedup as compared to the CPU-based system at less than 0.01% in real time. In [10], the MFCC process was implemented on the Xilinx Virtex-II XC2VP100

Received 21 April 2015, **Revised** 18 May 2015, **Accepted** 24 June 2015
***Corresponding Author** Il-Young Moon (E-mail: iymoon@koreatech.ac.kr, Tel: +82-41-560-1493)
School of Computer Science & Engineering, Korea University of Technology, 1600 Chungjeol-ro, Byeongcheon-myeon, Dongnam-gu, Cheonan 31253, Korea.

Open Access http://dx.doi.org/10.6109/jicce.2015.13.3.145 print ISSN: 2234-8255 online ISSN: 2234-8883

FPGA platform, which demonstrates a 150× speedup as compared to a CPU-based system at less than 0.09% in real time. In [5, 14], researchers attempted to optimize the ASR system on an ARM-based platform, which has been studied extensively because of the considerable increase in the use of mobile devices.

According to previous studies [4-6, 10], a highly parallelized and optimized MFCC architecture on the FPGA platform can improve the execution time of an automatic speech recognition (ASR) system as compared to the CPU.

In this paper, we propose a highly parallelized and optimized MFCC architecture implemented on the Xilinx Zynq-7000 system on a chip (SoC) platform and demonstrate that it is considerably faster than the CPU and the ARM processor. A C-based MFCC algorithm is executed on the CPU and the ARM, and the Verilog HDL MFCC algorithm is implemented on FPGA [13].

The rest of this paper is composed of the following four sections: background, design description, analysis and verification of results and performance, and conclusion.

II. BACKGROUND

Feature extraction is a process that extracts valid feature parameters from an input speech signal. Even if the same word is spoken, no two speech instances can produce the same speech waveform. The reason for this phenomenon is that the speech waveform includes not only speech information but also the emotional state and tone of the speaker. Therefore, the goal of speech feature extraction is to extract feature parameters that represent speech information. Further, this process is a part of compressing speech signals and modeling the human vocal tract. The feature parameters are devised to represent the phonemes accurately for speech recognition. Linear predictive coefficients (LPCs) and MFCCs are commonly used for the abovementioned feature extraction [3].

A. LPC Feature Extraction

LPC feature extraction starts with attempts to predict the value of the current sample from the total sum of a certain number of past samples multiplied with certain coefficients. The coefficients are called LPCs when in terms of the transfer function, the coefficients are formed in an electrode model (all-pole). Each polarity represents the position of the resonance frequency in the frequency domain and the transfer function of the vocal tract in the form of a spectral envelope approximation. For extracting the LPCs, the Levinson–Durbin algorithm was developed; it obtains the autocorrelation for a segment of speech and efficiently computes the LPCs by using a recursive method [1, 3].

B. MFCC Feature Extraction

MFCC feature extraction is a cepstral coefficient extraction method that reflects the characteristics of hearing. The aspect of the human ear responding to a frequency change is not linear but in the mel scale, which is similar to the logarithmic scale. According to the mel scale, a low frequency is sensitive to small changes, but the sensitivity decreases with an increase in the frequency. Therefore, MFCC is a correlation method performed during the frequency analysis step of the feature extraction [8].

1) Pre-emphasis

The input speech signal goes through a pre-emphasis filter, which has high-pass filter characteristics. The reason for using this high-pass filter is to model the frequency characteristics of the human external ear and middle ear. The high-pass filter compensates the attenuation by 20 dB/dec of the speech signal from the lips in order to obtain the vocal tract characteristics. Further, the high-pass filter compensates for the fact that the human auditory system is sensitive in the spectral region over 1 kHz. Once the input speech signal goes through a pre-emphasis filter, low-frequency values decline but high-frequency values get emphasized and boost the vocal tract characteristics. The pre-emphasis filter can be expressed by the following equation:

$$Y[n] = X[n] - aX[n-1],$$

where a denotes the pre-emphasis coefficient, n the time, and X the input data.

2) Frame Blocking and Hamming Windowing

After the pre-emphasis process, the input speech signal is divided into frame blocks of 16 ms in order to extract the feature parameters of the signal. The reason for dividing the input signal into frames of 16 ms is that the human voice has a stationary feature in a 16-ms frame. After dividing the input signal into frames, we extract the frequency feature of each frame. At the edges of each frame, there are discontinuities in the input signal that contain unnecessary information. In order to minimize the discontinuities at the edges of the frames, each frame is multiplied with the window coefficients, as shown in Fig. 1. For the window process, we can use the Hanning, Hamming, Blackman, and Kaiser methods. In this study, we applied the commonly used Hamming window method as follows:

$$W_H(n) = 0.54 - 0.46\cos(\tfrac{2\pi n}{N-1}).$$

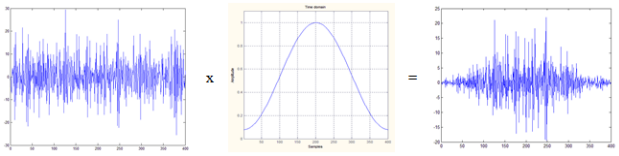

Fig. 1. Hamming window process.

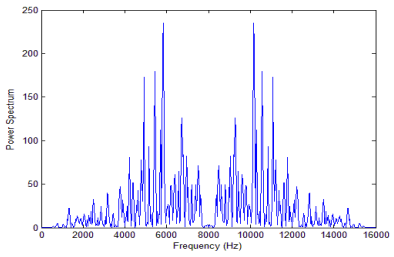

Fig. 2. Power spectrum results.

3) Fast Fourier Transform (FFT)

In order to extract the feature parameters of the input speech, the FFT algorithm can be applied to convert the time domain into the frequency domain to figure out the frequency characteristics of the input. In the time domain, a speech signal has discrete non-periodic features. Through a FFT, which converts the time domain into the frequency domain, a speech signal is transformed into a continuous periodic signal. The FFT algorithm is an efficient and fast algorithm for executing a discrete Fourier transform (DFT) and its inverse transform.

The N-point DFT equation of a sequence $x(n)$ can be described as follows:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}.$$

In the above equation, $e^{-j2\pi\frac{kn}{N}}$ rotates clockwise along the kn value in a complex plane. Because of the rotating feature, $e^{-j2\pi\frac{kn}{N}}$ is called the twiddle factor and is represented as W_N^{kn}.

4) Power Spectrum

In order to calculate the magnitude of the output of the FFT to emphasize a specific frequency feature, the energy spectrum is estimated. The energy spectrum is found to be real and symmetric (see Fig. 2). Because of its symmetric properties, we can use only half of the output points in the next step, and doing so helps to reduce the computational complexity.

$$\frac{1}{N}|S(m,k)|^2 = \frac{1}{N}(S_r^2(m,k) + S_i^2(m,k)),$$

where m represents the frame index and k indicates the frequency index (k = 0, 1, …, N – 1).

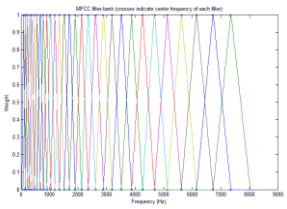

Fig. 3. Mel-filter bank.

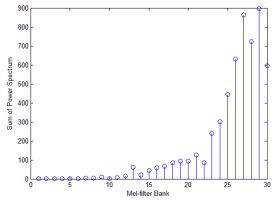

Fig. 4. Mel-filter coefficients.

5) Mel-Filter Bank

The human ear responds non-linearly to a speech signal. When the speech recognition system performs a non-linear process, it improves the recognition performance. By applying a mel-filter bank, we can obtain a non-linear frequency resolution. The mel-filter bank method is widely used in the speech recognition process.

As shown in Fig. 3, the mel-filter bank has a triangular shape and is applied to the output of the energy spectrum. The number of items in a mel-filter bank set is normally between 20 and 40. In this study, we use 19 mel-filter banks. These mel-filter banks are placed on the frequency axis on the basis of the mel scale, which is defined below. In order to calculate the energy of each mel-filter bank, the output of the energy spectrum is multiplied by the mel-filter bank coefficients and accumulated. By applying the mel-filter bank, we obtain 30 mel-filtered energy coefficients to ensure useful signal energy, as shown in Fig. 4.

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

6) Mel Cepstrum

Mel cepstrum is the final output of the MFCC process. The logarithm and discrete cosine transform (DCT) of the mel-filter bank energy are computed to extract the required minimum information. The reason why the log value of the mel-filter energy is taken is that the human ear responds to the loudness of the sound as a function of the logarithm (see Fig. 5). In the next step, DCT is applied to the log filter bank parameters in order to extract the appropriate features. The DCT equation is defined as follows:

$$c(i) = \sqrt{\frac{2}{L}} \sum_{m=1}^{L} log(\tilde{s}(m)) \cos\left[\frac{\pi i}{L}(m - 0.5)\right], i = 0,1,\ldots,C-1$$

where L denotes the number of filter banks (see Fig. 6).

Eventually, the MFFCs are obtained through all the steps, as shown in Fig. 7.

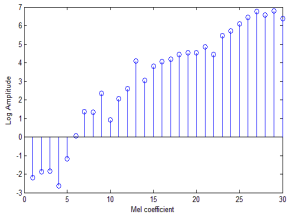

Fig. 5. Logarithm coefficients.

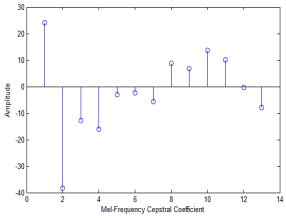

Fig. 6. DCT coefficients.

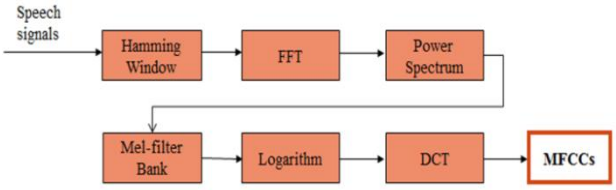

Fig. 7. Feature extraction process.

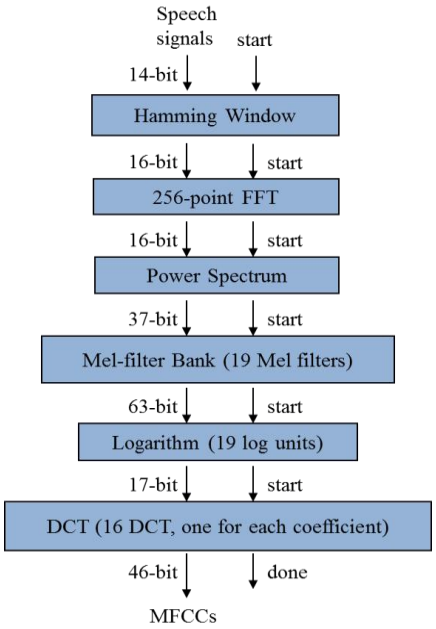

Fig. 8. Block diagram of the MFCC processing unit.

III. DESIGN DESCRIPTION

In this study, the MFCC architecture has been designed to be parallelized and optimized on the Xilinx Zynq-7000 SoC platform in order to speed up the real-time speech recognition. Further, in order to improve the execution time of the MFCC process, a well-designed FFT algorithm was developed as part of Carnegie Mellon University (CMU)'s Spiral Project [15]. The C-based MFCC architecture for the CPU and ARM experiment was developed as part of Massachusetts Institute of Technology (MIT)'s feature extraction project [13].

In order to simulate and implement the feature extraction process on CPU, ARM, and FPGA, MATLAB v7.11 (R2020b), Microsoft Visual Studio Express 2013, and Xilinx Vivado Design Suite (v2013.2), Integrated Software Environment (ISE), Software Development Kit (SDK), and High-Level Synthesis (HLS) tools were employed.

The objective of this study is to determine the improvement of the speech recognition system in terms of speed by implementing a parallel MFCC process on FPGAs to perform the feature extraction process.

A. MFCC Simulation on CPU and ARM

As mentioned above, the C-based MFCC architecture was developed as part of MIT's feature extraction project. For this experiment, a 6-s Wall Street Journal wave file was used as the MFCC input voice. The sample frequency of the speech signal was 16000 Hz. In order to extract the frequency feature, the speech signal was divided into 16-ms samples. When the sample frequency was 16000 Hz, we obtained 256 16-ms samples (0.016 s × 16000 Hz = 256 samples). The 16-ms signal block had a 10-ms overlap with the next 16-ms signal block because the signal overlap was required to recover the discontinuity of the signal. However, when the overlap was applied to the signal, signal distortion occurred. To prevent the signal distortion due to the signal overlap, the hamming window module was applied. In order to apply the MFCC process to the 6-s Wall Street Journal waveform speech signal with the signal overlap, the speech signal was divided into 1025 frames (6 s × 16000 Hz = 104711 samples, 0.016 s × 16000 Hz = 256 samples, (1 + ((104711 – 256)/100) = 1044.55)). The CPU experiment was conducted using Microsoft Visual Studio, and the ARM experiment was performed using Xilinx SDK.

B. MFCC Simulation on FPGAs

The MFCC process was implemented in the Verilog hardware description language (HDL); it included the hamming window, FFT, power spectrum, mel-filter, logarithm, and DCT processes. The FPGA experiment was executed in the Xilinx ISE tool.

Fig. 8 describes the MFCC process. Overall, the MFCC process consisted of 39 distinct modules (1 hamming window, 1 FFT, 1 power convertor, 19 mel-filters, 1 log calculator, and 16 DCTs) and 57 total module instances (1 hamming window, 1 FFT, 1 power convertor, 19 mel-filters, 19 log calculators, and 16 DCTs). As mentioned before, in order to improve the execution time of the MFCC process, a well-designed 16-bit 256-point FFT algorithm was developed

as part of CMU's Spiral Project. For a highly parallelized and optimized MFCC structure, the mel-filter, log calculator, and DCT modules were implemented as a parallel structure.

IV. ANALYSIS AND VERIFICATION OF RESULTS AND PERFORMANCE

We verified the speed improvement of the MFCC process on FPGA as compared to the CPU and ARM process. For the test, the 6-s Wall Street Journal wave file was used, which consisted of 1025 frames. The result was analyzed by comparing the MFCC execution time of one frame.

A. Experimental Setup

The experiment uses Intel Core i5 M 480 CPU at 2.67 GHz, Dual ARM Cortex-A9 MPCore at 667 MHz, and Zynq-7000 FPGA at 111 MHz. In order to simulate and evaluate the MFCC process time on the CPU, ARM, and FPGA, we used Microsoft Visual Studio Express 2013 for the CPU evaluation, Xilinx SDK v2013.4 for the ARM evaluation, and the Xilinx ISE v14.7 tool for the FPGA evaluation.

B. Analysis of Feature Extraction

For the analysis, the execution time per frame in the MFCC process was compared among the CPU, ARM, and FPGA. The MFCC process was divided into the following five steps: hamming window, 256-point FFT, power convertor, mel-filter and log convertor, and DCT.

Table 1. Average elapsed time per frame in MFCC process

	CPU (μs)	ARM (μs)	FPGA (μs)
Hamming	30.688	0.006	1.415
FFT	2164.624	148.759	1.521
Power	4.973	2.669	1.414
Mel-filter and log	3.324	0.500	0.0438
DCT	8.305	13.122	0.0219
Total	2211.915	165.055	4.415

Table 2. Relative speedup of MFCC process compared to CPU

	CPU (μs)	FPGA (μs)	Relative speedup
Hamming	30.688	1.415	21×
FFT	2164.624	1.521	1423×
Power	4.973	1.414	3×
Mel-filter and log	3.324	0.0438	75×
DCT	8.305	0.0219	379×
Total	2211.915	4.415	501×

Table 3. Relative speedup of MFCC process compared to CPU

	ARM (μs)	FPGA (μs)	Relative speedup
Hamming	0.006	1.415	0.004×
FFT	148.759	1.521	97×
Power	2.669	1.414	1.8×
Mel-filter and log	0.500	0.0438	11×
DCT	13.122	0.0219	599×
Total	165.055	4.415	37×

Table 1 denotes the execution time on each platform in micro-seconds. Tables 2 and 3 describe the relative speedup of the MFCC process on FPGA compared to that on the CPU and ARM platforms.

Through the analysis, we confirmed that the FPGA platform is approximately 500× faster than a sequential CPU platform and 60× faster than a sequential ARM platform, and verified that a highly parallelized and optimized MFCC architecture on the FPGA platform significantly improves the execution time of an ASR system as compared to the CPU and ARM platforms. In order to improve the execution time of the MFCC process on FPGA, a well-designed 256-point FFT algorithm was developed as part of CMU's Spiral Project [15]. The C-based MFCC architecture for the CPU and ARM platforms was developed as part of MIT's feature extraction project [6].

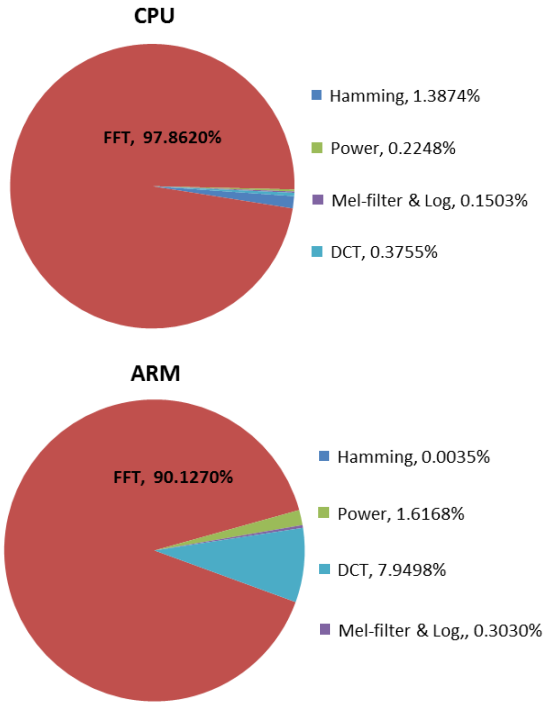

Fig. 9. Percentage of time spent on each processing stage in the MFCC process on the CPU and ARM.

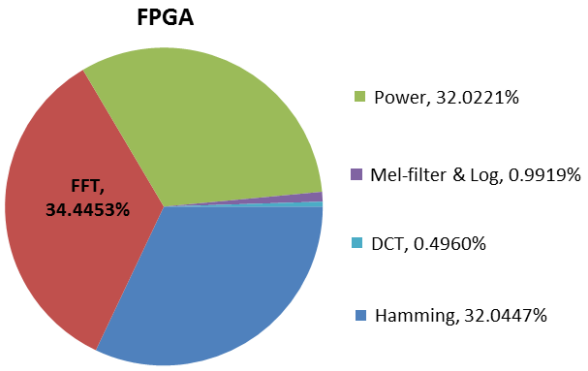

Fig. 10. Percentage of time spent on each processing stage in the MFCC process on FPGA.

As seen in Fig. 9, the FFT process consumes a large amount of computational time on the CPU and ARM. CPU and ARM consume 97.8% and 90.1% of the execution time, respectively, in the MFCC process. On the other hand, as shown in Fig. 10, on the FPGA platform, the FFT process consumes 34.4% of the execution time in the MFCC process and significantly reduces the computational time compared to CPU and ARM.

V. CONCLUSION

The objective of this study was to determine the improvement of the speech recognition system in terms of speed by implementing a parallel feature extraction process on FPGA for feature extraction. The Xilinx Zynq-7000 SoC platform was used for demonstrating the MFCC implementation for the feature extraction process. We confirmed that the FPGA platform is approximately 500× faster than a sequential CPU platform and 60× faster than a sequential ARM platform, and verified that a highly parallelized and optimized MFCC architecture on the FPGA platform significantly improves the execution time of an ASR system compared to the CPU and ARM platforms.

ACKNOWLEDGMENTS

We would like to thank Xilinx Inc. for providing the FPGA hardware and software tools developed as part of Xilinx University Program. We would also like to thank Won-Kyum Lee, a Ph.D. candidate at CMU, and In-Hwan Kim, an M.S. candidate at SJSU, for their support throughout this research.

REFERENCES

[1] S. T. Pan, C. F. Chen, and J. H. Zeng, "Speech recognition via Hidden Markov Model and neural network trained by genetic algorithm," in *Proceedings of International Conference on Machine Learning and Cybernetics (ICMLC)*, Qingdao, China, pp. 2950-2955, 2010.

[2] M. Mohri, F. Pereira, and M. Riley, "Speech recognition with weighted finite-state transducers," in *Springer Handbook of Speech Processing*. Berlin: Springer, pp. 559-584, 2008.

[3] J. G. Kim, H. Y. Junh, and H. Y. Chung, "The improvement of the Korean Speech recognition systems using MEL-LPC analysis method," *Journal of the Institute of Information and Telecommunication*, vol. 9, no. 1, pp. 65-70, 2002.

[4] K. You, J. Chong, Y. Yi, E. Gonina, C. J. Hughes, Y. K. Chen, W. Sung, and K. Keutzer, "Parallel scalability in speech recognition," *IEEE Signal Processing Magazine*, vol. 26, no. 6, pp. 124-135, 2009.

[5] D. Huggins-Daines and A. Rudnicky, "Mixture pruning and roughening for scalable acoustic models," in *Proceedings of ACL Workshop on Mobile Language Technologies*, Columbus, OH, pp. 21-24, 2008.

[6] H. Kou, W. Shang, I. Lane, and J. Chong, "Optimized MFCC feature extraction on GPU," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, pp. 7130-7134, 2013.

[7] M. Staworko and M. Rawski, "FPGA implementation of feature extraction algorithm for speaker verification," in *Proceedings of the 17th International Conference on Mixed Design of Integrated Circuits and Systems (MIXDES)*, Warsaw, Poland, pp. 557-561, 2010.

[8] J. C. Wang, J. F. Wang, and Y. S. Weng, "Chip design of MFCC extraction for speech recognition," *Integration, the VLSI Journal*, vol. 32, no. 1, pp. 111-131, 2002.

[9] M. Bahoura and H. Ezzaidi, "Hardware implementation of MFCC feature extraction for respiratory sounds analysis," in *Proceedings of 8th Workshop on Systems, Signal Processing and their Applications (WoSSPA)*, Algeria, pp. 226-229, 2013.

[10] E. M. Schmidt, K. West, and Y. E. Kim, "Efficient acoustic feature extraction for music information retrieval using programmable gate arrays," in *Proceedings of 10th International Society for Music Information Retrieval Conference (ISMIR2009)*, Kobe, Japan, pp. 273-278, 2009.

[11] S. Ke, Y. Hou, Z. Huang, and H. Li, "A HMM speech recognition system based on FPGA," in *Proceedings of Congress on Image and Signal Processing (CISP2008)*, Sanya, China, pp. 305-309, 2008.

[12] K. You, H. Lim, and W. Sung, "Architecture design and implementation of an FPGA softcore based speech recognition system," in *Proceedings of IEEE International Workshop on System-on-Chip for Real-Time Application (IWSOC)*, Cairo, Egypt, pp. 50-55, 2006.

[13] MFCC project: c-based algorithm of MFCC [Internet], Available:

https://code.google.com/p/mfcc-umbc/wiki/MFCCIntro.

[14] D. Huggins-Daines, M. Kumar, A. Chan, A. Black, M. Ravishankar, and A. Rudnicky, "PocketSphinx: a free, real-time continuous speech recognition system for hand-held devices," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2006)*, Toulouse, France, 2006.

[15] Spiral project: DFT/FFT IP core generator [Internet]. Available: http://www.spiral.net/hardware/dftgen.html.

Chang Choo

received the B.S. and M.S. degrees in industrial engineering from Seoul National University, Korea, in 1977 and 1981, respectively, and the M.S. and Ph.D. degrees in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, New York, USA, in 1982 and 1986, respectively. From 1979 to 1981, he was a Researcher at the Korea Advanced Energy Research institute, and from 1986 to 1991 he was an Assistant Professor in the Department of Electrical Engineering, Worcester Polytechnic Institute, Worcester, Massachusetts, USA. Since 1991, he has been on the faculty with the Electrical Engineering Department at California State University, San Jose. He was with several Silicon Valley companies, including Altera, National Semiconductor, and Philips Semiconductor. His current research interests include embedded SoC design, DSP architecture and system, and digital image/audio processing. Dr. Choo is a member of the IEEE, SPIE, ASEE, and Eta Kappa Nu.

Young-Uk Chang

received the BS degree in elecrtical engineering from Hanyang University, Seoul, Korea in 2010, and the MS degree in electrical engineering from San Jose State University, San Jose, California, USA in 2013. Since 2014, he has been a display IC design engineer with Siliconworks, Daejeon, Korea.

Il-Young Moon

received the B. S., M. S. and Ph. D. degrees in telecommunication and information engineering from Hankuk Aviation University, Kyounggi-Do, Korea in 2000 and 2002 and 2005 respectively. He is currently a associate professor of School of Computer Science & Engineering, Korea University of Technology and Education, Korea. His research interests are in the areas of mobile IP, wireless LANs, wireless TCP, Ad hoc Network. He is a member of Korean Institute of Communication Sciences, a member of Korea Institute of Electronics Engineers and a member of Korea Navigation Institute.