Design and Implementation of a SoPC System for Speech Recognition

Article *in* Lecture Notes in Electrical Engineering · January 2013

DOI: 10.1007/978-94-007-6738-6_147

4 authors, including:

Tran Van Hoang
Ho Chi Minh City University of Technology (H…
4 PUBLICATIONS **1** CITATION
SEE PROFILE

Nguyen LY Thien Truong
Ho Chi Minh City University of Technology (H…
1 PUBLICATION **1** CITATION
SEE PROFILE

Xuan-Tu Tran
Vietnam National University, Hanoi
41 PUBLICATIONS **173** CITATIONS
SEE PROFILE

Design and Implementation of a SoPC System for Speech Recognition

Tran Van Hoang[1], Nguyen Ly Thien Truong[1], Hoang Trang[1], Xuan-Tu Tran[2]

[1] University of Technology, Vietnam National University, HoChiMinh City.
{tvhoang, nlttruong, hoangtrang}@hcmut.edu.vn
[2] VNU University of Engineering and Technology – 144 Xuan Thuy, Cau Giay, Hanoi.
tutx@vnu.edu.vn

Abstract. This paper presents the design of a System on Programmable Chip (SoPC) based on Field Programmable Gate Array (FPGA) for speech recognition in which Mel-Frequency Cepstral Coefficients (MFCC) for speech feature extraction and Vector Quantization for recognition are used. The implementing process of the Vietnamese recognition system undergoes the following steps: feature extraction, training codebook, recognition. In the first step, feature extraction, the input voice data will be transformed into spectral components and extracted to get the main features by using MFCC algorithm. In the recognition step, the obtained spectral features from the first step will be processed and compared with the trained components from the second step. To be easily implemented on FPGA, the Vector Quantization (VQ) is applied in this step. In our experimental, Altera's DE2 board with Cyclone II FPGA is used to implement the recognition system which can recognize 64 words. The feature extraction, recognition is implemented on SoPC. The training of codebook is implemented on PC using C/C++ program. The execution speed of the blocks in the speech recognition system is surveyed by calculating the number of clock cycles while executing each block.

Keywords. Speech recognition, MFCC, VQ, SoPC, FPGA, Nios

1 Introduction

Speech recognition system is applied in many application fields such as health care, military, human computer interaction, avionics technicians…[1], especially, the applications which support disabled people to communicate with the world in a better way. For that reason, there are many studies on software/hardware implementation of speech recognition systems for many years. However, because of a large number of accents spoken around the world, there are still many challenges that need further research and development, for example, Vietnamese speech recognition.

The research on speech recognition going mainly in two directions, namely: the software runs on Personal Computers (PCs) and embedded systems. For the first direction, there exit many studies and software tools, which are developed successfully. In particular, the Hidden Markov Model Toolkit (HTK) is a toolkit for building Hidden Markov Models (HMMs) used in speech recognition successfully [2]. There are also many tools running on the PC or smart phone aimed at the control device via speech. For the second direction, embedded systems have many advantages as high performance, convenience, low cost, and great development potential. However, speech recognition research based on embedded systems is more difficult. This paper will present the implementation of a speech recognition system as an embedded system using FPGA technology.

In fact, the implementation of speech recognition systems has been done using FPGA technology in recent years. In paper [3], speech recognition systems are implemented as hardware/software co-design systems using Hidden Markov Model (HMM). This project use Linear Predictive Coding (LPC) method in feature extraction block. So, the recognition accuracy is not high compared with the MFCC method. In paper [4], the MFCC method is applied, but the optimization was not taken into account yet to increase performance. Another work, presented in [1] and [1], the author proposed an efficient MFCC hardware implementation for feature extraction in speech recognition. However, this work has been done using ASIC technology and therefore less flexible than FPGA based implementations. Other implementations for speech recognition systems can be found at [7-10]. Among these, the work presented in [8] proposes a hardware/software co-design method to tradeoff between the performance and the flexibility of the recognition system while [7] and [10] present FPGA based implementation of the recognition systems. None of them discuss about the optimization method for MFCC algorithm. In our work, the MFCC method is used with some modifications to increase the performance of the system. The whole system has been implemented using Altera FPGA technology to be more flexible.

The block diagram of speech recognition system is shown in Fig. 1. Audio samples go through feature extraction block to retrieve the characteristics of sound. Through the Feature Extraction block, the audio input will be transformed into the spectral coefficients. Then, these spectral characteristics go through Training block to create codebook for each word. In the recognition step, Recognition block uses spectral features and compares with the codebooks which are trained above.

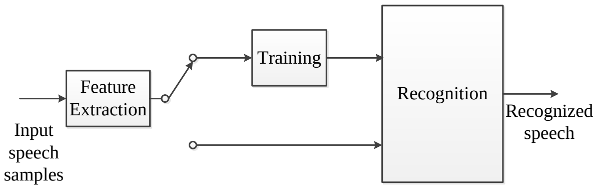

Fig. 1. Speech recognition system.

The paper is organized as follows. The theory of speech recognition is briefly presented in Section 2. The design and implementation of the proposed speech recognition system as a SoPC (System on Programmable Chip) is mentioned and discussed in Section 3. Section 4 will show the achieved experimental results. Finally, conclusions and further discussions will be presented in Section 5.

2 Overview of Speech Recognition

In this section, the overview of speech recognition will be presented in a fundamental way, which is mainly present the theory related to the implementation of our practical system. Major contents include the following: the algorithm of feature extraction is presented in Section 2.1. Vector quantization technique is presented in Section 2.2.

2.1 Feature extraction

In speech recognition, voice feature extraction is first step we need to make. Feature extraction process gives the parameters used for recognition stage easily than the original speech signal. One of the most efficient algorithms used in feature extraction is MFCC algorithm.

This method is based on the perceived sound of the human ear that is linear in the low frequencies and increases with logarithmic scales in the high frequencies. From this characteristic, the MFCC method gives us the most important characteristics of the human voice. Fig. 2 presents the conventional MFCC algorithm for feature extraction.

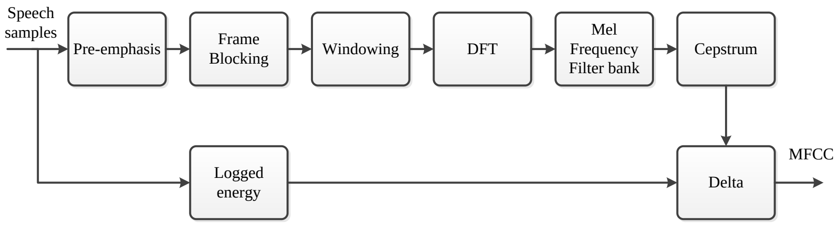

Fig. 2. Conventional MFCC feature extraction algorithm.

2.2 Vector Quantization process

The Vector Quantization process is described in Fig. 3. Audio signal after being extracted features will produce a series of feature vectors.

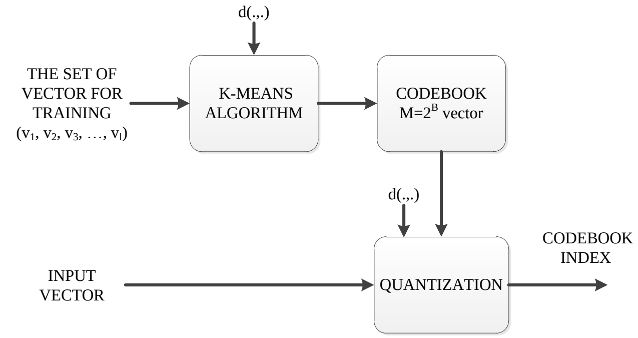

Fig. 3. Vector Quantization (VQ) training and classification.

Then, feature vectors will be quantized to be split into different M groups are called codebook ($M =$ 16/32/64/128/...) and each group will be labeled from 1 to M.

In speech recognition, it is common to use the Euclidean distance:

$$d(x, y) = \sqrt{\sum_{k=1}^{P} |x_k - y_k|^2} \tag{1}$$

This measure used in the classification stage, labeled feature vector. In addition, it is used in the recognition step. K-Means algorithm is used for codebook generation.

3 Implementation

In the implementation process, some blocks will be adjusted, modified so that the computing speed of the block can be increased. In this section, we will show some improvements in a few blocks to optimize the computing speed of the block. Evaluated results in terms of the number of clock cycles will be presented in the next section.

3.1 Feature Extraction implementation

Voice Activation Detection (VAD)

Voice signal after recording through the microphone will gain a certain number of samples. In this project, the sampling frequency is $8kHz$, each time recording in 1 second, corresponding to 8000 samples. However, in the 8000 samples, not all are meaningful sound, much of them are silence. So, before the audio samples are extracted features, it requires the program to extract the significant audio and remove the silence.

As mentioned in previous section, audio signal is divided into M segments (i.e., blocks), L samples in each segment. In this project, we assigned that $L = 160$ with $F_s = 8000Hz$, which means $20ms$ for each segment.

Then the energy function E_s will be calculated for each segment by the following formula:

$$E_s(m) = \sum_{n=L*m+1}^{L*(m+1)} s_1^2(n) \tag{2}$$

VAD will reject segment m if $E_s(m) <$ TH. In this project, TH $= 0.0001$. The selection of TH is due to the test, go back and forth several times to select the appropriate value makes the correct signal clipping.

Pre-emphasis

In pre-emphasis block, the coefficient "a" has the value from 0.9 to 1. In theory, the normal value of "a" is 0.97. But, when we build the system on SoPC, we must choose the value of "a" so that the program easy to implement. Thus, the pre-emphasis block will run faster. The value $a =$ 1, 15/16, 0.97 are surveyed about program performance through assessment of pulse clock.

Transfer function of the filter is described by Equation 3. In the time domain, the relationship between output and input is shown in Equation 4.

$$H(z) = 1 - a.z^{-1} \tag{3}$$

$$s_i' = s_i - a.s_{i-1} \tag{4}$$

With $a = 1$, Equation 4 will be simplified as: $s_i' = s_i - s_{i-1}$.

Advantage of using 15/16 as "a" coefficient is expressed in Equation 5. $\frac{15}{16}s_{i-1}$ can be realized in binary computation system by shifting s_{i-1} 4 bits to the right. Using this value the multiplication step is simplified to shift and subtract operations.

$$s_i' = s_i - a.s_{i-1}, \quad a = \frac{15}{16} \tag{5}$$

$$s_i' = s_i - \frac{15}{16}s_{i-1} = s_i - (s_{i-1} - \frac{1}{16}s_{i-1}).$$

Discrete Fourier Transform (DFT)

In general, $X(k)$ and $x(n)$ are the complex numbers. N-point DFT can be calculated as follows:

$$X_R(k) = \sum_{n=0}^{N-1}\left[x_R(n)cos\frac{2\pi kn}{N} + x_I(n)sin\frac{2\pi kn}{N}\right], \quad k = 0, 1, 2, \ldots, N-1. \tag{6}$$

$$X_I(k) = -\sum_{n=0}^{N-1}\left[x_R(n)sin\frac{2\pi kn}{N} - x_I(n)cos\frac{2\pi kn}{N}\right], \; k = 0, 1, 2, \ldots, N-1. \tag{7}$$

If DFT transformation uses two equations 6 and 7 to calculate, it costs $2N^2$ trigonometric calculations, $4N^2$ real multiplications, and $4N(N-1)$ additions. This shows that when the direct calculation using the DFT formula above arises large computational cost, it will slow speed program execution. Therefore, in this case we use the Fast Fourier Transform (FFT) algorithm instead. In addition, by using the look-up table of coefficients cosine, sine also increases the computing speed of the program.

Magnitude computation

If using the conventional formula for calculating the complex amplitude as Equation 8, then the calculation will be very slow speed, thereby reducing the speed of program execution.

$$M = \sqrt{I^2 + Q^2} \tag{8}$$

Therefore, the estimation algorithm is applied. This algorithm calculates very fast amplitude of a complex number almost exact compared to the normal range by taking the square root operation. For complex number $I + jQ$, amplitude estimation algorithm as follows:

$$M \approx \alpha.\max\{|I|, |Q|\} + \beta.\min\{|I|, |Q|\} \tag{9}$$

In this system, we use α as 1 and β as 1/4. This approach reduces the number of calculations with acceptable error.

Mel frequency filter bank

The $k^{th}$of power coefficient of the n^{th} frame is calculated by the Equation 10 as

$$S'_{nk} = \sum_j S_{nj}.FC_{kj}, \quad k = 0, 1, \dots, K \tag{10}$$

where, K is the number of the filters. S_{nj} is the j^{th} point of the n^{th} frame's spectrum, and FC_{kj} is the j^{th} coefficient of the k^{th} filter. When implementing the speech recognition system on SoPC, the rectangular filter bank is used in the new algorithm instead of the triangular filter bank. So, the Equation 10 becomes

$$S'_{nk} = \sum_j S_{nj}.FC_{kj}, \quad FC_{kj} = 0 \text{ or } 1 \tag{11}$$

The rectangular filters are proposed to be used instead of the triangular filters because the output characteristic of a rectangular filter is either a "1" or a "0", the multiply and sum operations can be simplified to simple "add" and "no add" operations. No multiplication step is required in the proposed approach.

3.2 Vector Quantization implementation

In this project, codebook size of 128 is considered. We use K-Mean algorithm for training codebook. First, randomly choose M vectors in the L vectors for training. The second step, for each training vector v, we find the codeword in the current codebook vectors closest distance this vector and we assign it belongs to the group of the codeword. The third step, for each group, we update codeword using the focus of all training vectors in this group. Repeat steps 2 and 3 until quantum error smaller than threshold value. The algorithm to implement this scheme is illustrated in Fig. 4.

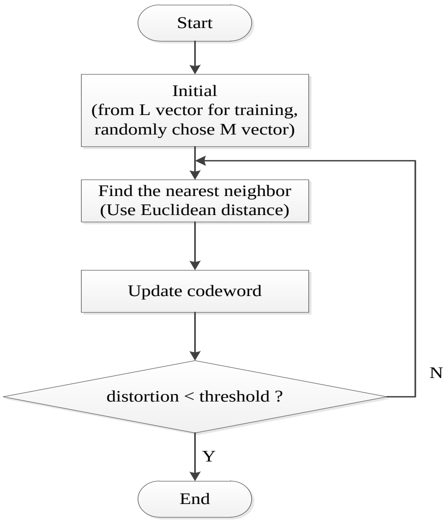

Fig. 4. Flowchart of the training codebook.

The input speech sample is extracted the feature by the MFCC algorithm first. Then the feature vectors are calculated to find the VQ distortion for each codebook. The word having smallest distortion is the word which needs to be identified (Fig. 5).

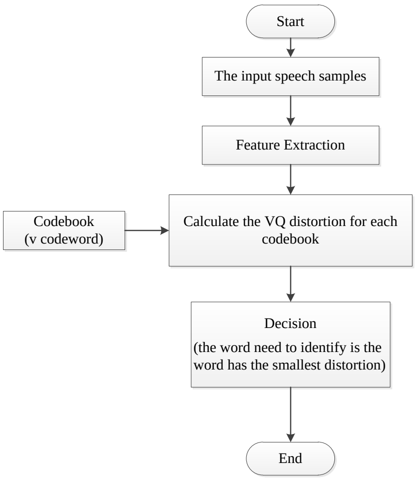
Start
The input speech samples
Feature Extraction
Codebook
(v codeword)
Calculate the VQ distortion for each codebook
Decision
(the word need to identify is the word has the smallest distortion)
End

Fig. 5. Flowchart of recognition.

3.3 SoPC implementation

The proposed speech recognition has been intently implemented on Altera FPGAs for high performance. To do that, we proposed a SoPC architecture for speech recognition system as described in Fig. 6.

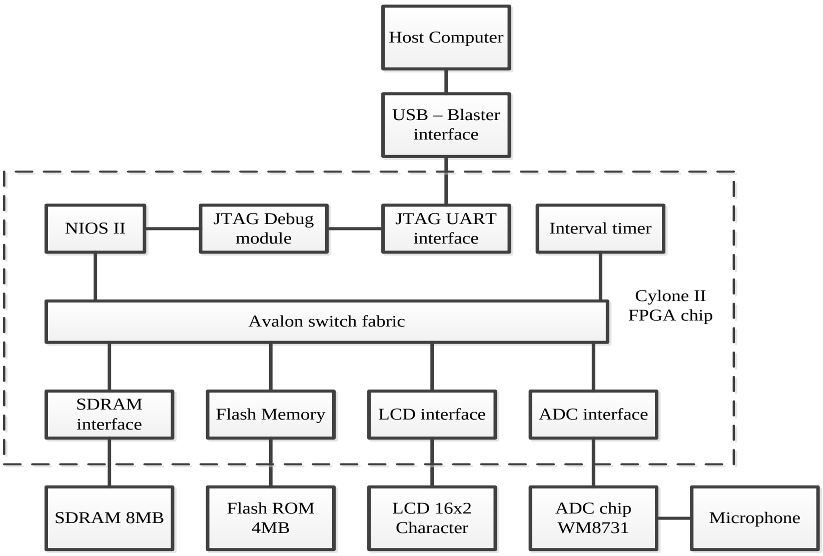

Fig. 6. Design of SoPC based on FPGA.

In this architecture, Nios II Processor is the most important component of the system, a processor to execute programs of the system. All compiled C program is stored in the SDRAM. Flash memory is used to store the parameters of the codebook after training. The ADC interface is the part connected to the Audio Codec WM8731 chip. This chip is responsible for data sampling of voices speak into the microphone. LCD is used to show the implementation of the program, the recognition results will be also displayed on LCD. In particular, the Interval Timer is used to calculate the number of the pulse clocks when executing each block.

4 Experimental Results

As mentioned above, we use Interval Timer to survey the program execution speed of each functional block in speech recognition system. The input speech samples are used for system input is 2400 samples. The clock of the system is $50MHz$.

4.1 Feature Extraction

First, the value $a = 1$, 15/16, 0.97 are surveyed about program implementation speed. From there will know the impact of the coefficient "a" on the performance speed of the pre-emphasis block. The obtained results are presented in Table 1.

Table 1. Obtained results of program execution speed by coefficient "a"

	$\boldsymbol{a = 1}$	$\boldsymbol{a}$ **=15/16**	$\boldsymbol{a = 0.97}$
Number of clock cycles to execute *pre_emphasis* block (*cycles*)	2,078,463	2,155,870	2,156,018

As we can see in Table 1, the pre-emphasis block with $a = 1$ is executed fastest. The value $a = 15/16$ in the pre-emphasis block run faster than the pre-emphasis block with $a = 0.97$.

In the Fourier transform step, DFT algorithm is replaced by the FFT algorithm to increase the speed of execution. As shown in the Table 2, FFT algorithm runs faster than DFT algorithm very much.

Table 2. Results of FFT and DFT

	FFT	**DFT**
Number of clock cycles to execute FFT/DFT block (*cycles*)	94,874,620	365,586,715

In the magnitude computation step, the estimation algorithm calculates very fast amplitude of a complex number almost exact compared to the normal algorithm by taking the square root operation, see Table 3.

Table 3. Results of magnitude computation

	Estimation amplitude	**Accuracy amplitude**
Number of clock cycles to execute magnitude_computation block (*cycles*)	7,463,640	80,412,716

By using the rectangle filters to replace the triangle filters, the program execution speed of the Mel Frequency Filter Bank block is increased 46 times, as in Table 4.

Table 4. Results of Mel frequency filter bank

	Rectangle filters	Triangle filters
Number of clock cycles to execute Mel-filter-bank block (*cycles*)	418,427	19,317,411

In Fig. 7, the program execution speed of all blocks in MFCC based feature extraction is shown. The FFT block is the slowest, requires 94,874,620 clock cycles to complete the given input samples. The Cepstrum block also costs many clock cycles because in this block the logarithm has not been optimized.

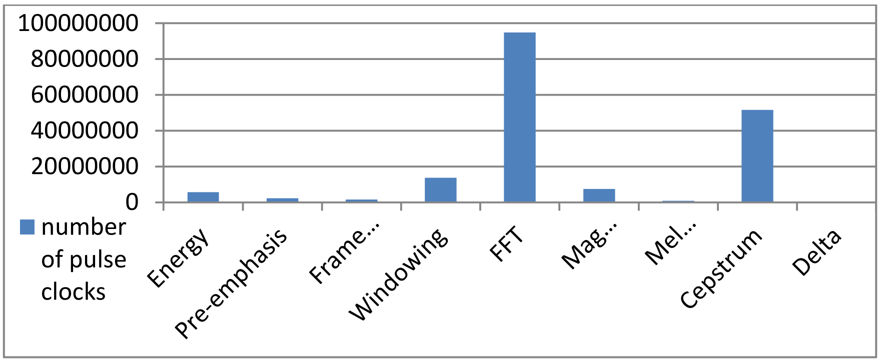

Fig. 7. Program execution speed of the blocks in MFCC based feature extraction.

4.2 Vector Quantization

With codebook size of 128, Vector Quantization is used in the recognition step. So, it costs 531,067,721 clock cycles.

4.3 Recognition accuracy

The whole recognition system with proposed architectures, parameters as stated above has the recognition accuracy of 88%, in which 7,416 utterances recorded from male and female adults in three regions of the North, Middle, and South of Vietnam are used.

5 Conclusion

In this paper, we propose efficient architectures and design choices for each part in MFCC-HMM-based speech recognition system to improve the processing speed. The determination of design choices are based on the easiness in implementation and experimental results of whole system. The whole system is built on FPGA, verified by testing with 7,416 utterances which are recorded from male and female adults in three regions of North, Middle and South of Vietnam with a recognition accuracy of 88%.

References

[1] Lawrance Rabiner & Biing – Hwang Juang: Fundamentals of Speech Recognition, Prentice Hall PTR, 1993.

[2] Thomas Hain, Gunnar Evermann, Dan Kershaw, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, Phil Woodland, Steve Young: The Hidden Markov Model Toolkit (HTK) Book (for HTK version 3.2.1), Cambridge University. Available at: http://htk.eng.cam.ac.uk/ (1995 - 2002).

[3] V. Amudha, B.Venkataramani, R. Vinoth kumar, S. Ravishankar: Software/Hardware Co-Design of HMM based Isolated Digit Recognition System. In: Journal of Computers, VOL. 4, No. 2, pp. 154-159, (2009).

[4] Haitao Zhou, Xiaojun Han: Design and Implementation of Speech Recognition System Based on Field Programmable Gate Array. In: Modern Applied Science, Vol. 3, No. 8, pp. 106-111, August 2009.

[5] Wei Han, Cheong-Fat Chan, Chiu-Sing Choy, Kong-Pang Pun: An Efficient MFCC Extraction Method in Speech Recognition. In: the 2006 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 145-148, Greece (2006).

[6] Wei Han: A Speech Recognition IC with an Efficient MFCC Extraction Algorithm and Multi-mixture Models, the Chinese University of Hong Kong, Doctor of philosophy thesis, September 2006.

[7] S.-T. Pan, C.-C. Lai and B.-Y. Tsai: The implementation of speech recognition systems on FPGA - based embedded systems with SOC architecture. In: International Journal of Innovative Computing, Information and Control, Volume 7, Number 10, October 2011.

[8] O. Cheng, W. Abdulla, Z. Salcic: Hardware-Software Co-design of Automatic Speech Recognition System for Embedded Real-Time Applications. In: IEEE Transactions on Industrial Electronics, pp. 850-859, March 2011.

[9] Weiqian Liang, Hui Geng: Design of speech recognition co-processor with fast Gaussian likelihood computation. In: the 3rd International Conference on Computer Research and Development (ICCRD), pp. 392-395, March 2011.

[10] Ge Zhang, Jinghua Yin, Qian Liu and Chao Yang: A real-time speech recognition system based on the Implementation of FPGA. In: Cross Strait Quad-Regional Radio Science and Wireless Technology Conference (CSQRWC), pp. 1375-1378, July 2011.