FPGA Implementation of Feature Extraction Algorithm for Speaker Verification

Michał Staworko
Institute of Telecommunications
Warsaw University of Technology
Warszawa, Poland
m.staworko@elka.pw.edu.pl

Mariusz Rawski
Institute of Telecommunications
Warsaw University of Technology
Warszawa, Poland
rawski@tele.pw.edu.pl

Abstract—**Speaker identification is a computationally demanding task. Particularly the stage of feature extraction, that is responsible for reducing the resources required to describe speech samples accurately requires algorithms of large complexity. State of the art speaker verification systems are based on cepstral features like MFCC, LFCC or LPCC. In these methods of features extraction the most computing extensive part is spectral averaging using windows in frequency domain. In this paper we propose a feature extraction system based on the LFCC with novel architecture for spectral averaging. Proposed solution is optimized for implementation in programmable structures as System on Programmable Chip and significantly reduces feature extraction execution time.**

Index Terms—**speaker identification; Linear Frequency Cepstral Coefficiens; FPGA; low power application**

I. Introduction

Speech is one of the most natural forms of communication. Recently, there have been a lot of research done to use speech as biometric feature in security systems. Biometry has an advantage over traditional method, since it eliminates such problems as lost, forgotten or stolen passwords (a biometric feature itself is a password). In speaker identification, the task is to use a speech sample to select the identity of the person that produced the speech from among a population of speakers. It is the most natural way of access protection. However, there is often a trade-off between ease of use and system complexity.

Biometric systems, are usually implemented on personal computers equipped with high-performance microprocessors. This is because of the computation complexity of applied algorithms, as well as their high confidential levels of security. General purpose processors contain floating-point units able to carry out millions of operations per second at frequencies in the GHz range, what allows to resolve the complex algorithms in just a few hundred of milliseconds. However, in the low-cost consumer market, such factors as price, power consumption and size determine the viability of a product. Since the main drawback of microprocessors based systems are the cost, and the necessary space required to incorporate their external associated peripherals, the use of an FPGA (*Field Programmable Gate Array*) becomes a suited way to implement systems that require a high computational capability at affordable prices. Additionally the FPGA allows to divide and implement algorithm as parallel parts, which allows to run computation at lower operational circuit frequency which requires less power consumption.

FPGA circuits can be programmed by the user and adapted to perform the particular task. Term "programming" in case of a FPGA architecture means changing its internal structure and can be repeated many times. This mechanism that allows for FPGA programming on the one hand decrease operating speed of FPGA chip comparing to ASIC, on the other hand it provides the possibility to tune-up the system to the specific parameters of implemented algorithm.

The speaker identification problem can be roughly divided into two issues: speech analysis (feature extraction) and classification. Feature extraction methods are responsible for reducing the resources required to describe speech samples accurately. In case of speech analysis various digital signal processing (DSP) algorithms are used to detect desired features of speech signal. Most common are Linear Prediction Coding (LPC), Mel-scale Filterbank Cepstrum Coefficients (MFCC), Linear-scale Filterbank Cepstral Coefficiens (LFCC) and others. MFCC is recognized as the best known and most popular. However LFCC algorithm is often used in speaker identification application, since it produces results of comparable quality [1], [2], [3].

There are some publications dealing with FPGA implementations of speaker verification systems. In [4], authors show the main features of an implementation of a SVM (*Support Vector Machines*) speaker verification system for Match-on-Card. The paper shows a FPGA implementation of the matching stage using a kernel based on an exponential function. The system is able to carry out the matching between the model and the feature vector 50 times faster than a software-based solution running on a Pentium IV clocked at 1.3 GHz. In [5] the implementation of a whole SVM speaker verification system based on dedicated hardware is presented. The system consists of several stages dedicated to calculate the feature vectors, based on mel-frequency cepstral coefficients. In [6] and [7] there are presented only hardware implementations of some specific part of algorithms for speech recognition or speaker identification, that allow a significant acceleration of the processing time.

This paper presents the feature extraction solution based on LFCC with the architecture specially optimized for implementation in FPGA structures. Discussed system is

Copyright © 2010 by Department of Microelectronics & Computer Science, Technical University of Lodz

designed to be implemented in FPGA device as System-on-Programmable-Chip (SOPC).

II. Feature Extraction Circuit

State of the art speaker verification systems are based on the estimation of the spectral envelope of the short term signal, e.g., Mel-scale Filterbank Cepstrum Coefficients, Linear-scale Filterbank Cepstrum Coefficients, or Linear Predictive Cepstrum Coefficients (LPCCs). All the features examined are based on spectral information derived from a short time-windowed segment of speech. They differ mainly in the detail of the power spectrum representation. The filterbank features (MFCC and LFCC) are derived directly from the FFT power spectrum, whereas the linear prediction-based features use an all-pole model to represent the smoothed spectrum. The mel-scale filter bank centers and bandwidths are fixed to follow the mel-frequency scale, giving more detail to the low frequencies, whereas the linear filter bank provides equal detail to all frequencies [1],[2],[3].

Linear Frequency Cepstral Coefficients (LFCC) are frequently used in speech and speaker recognition application and give similar results as MFCC, that is recognized as the state of [8],[9].

The block diagram of LFCC extractor is presented on the Fig. 1.

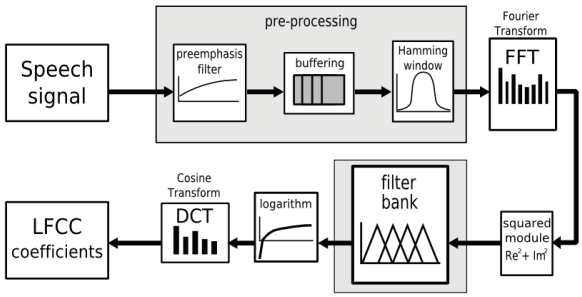

Figure 1. Schematic reprezetation of LFCC extractor

The process of computation of LFCC features for a given speech signal consists of the following steps:

- dividing speech signal into overlapping frames,
- compute the power spectrum of the signal frame,
- multiply and accumulate the power spectrum samples for each window from the filter bank,
- computation of the logarithm of obtained coefficients,
- compute the discrete cosine transform.

Performing computation on power spectrum allows to replace inverse Fourier transform with cosine transform. The cosine transform reduces computation to real values only, and additionally only the half of the power spectrum is taken to compute DCT. This significantly reduces the computation cost [8].

Methods used to compute the MFCC and LFCC parameters differ in filter bank that is applied for spectral averaging. To compute the MFCC the mel-scale [10] wrapped filter bank is used, in which windows located in lower frequency are far more slimmer than located in high freq, which gives precise description of low frequencies and rough image of high frequencies. This non linear mapping responds to human frequency perception [10].

To compute the LFCC parameters windows evenly distributed in frequency domain are used. That describes both, high and low frequency with the same precision. If comparing LFCC and MFCC filter bank the former gives better image of high frequency. In [11] it is concluded that a rich amount of speaker individual information is contained in the higher frequency band, and it is useful for speaker recognition. Thus LFCC may be consider as better speech signal estimators, since it provides better high-frequency resolution.

III. Spectral Averaging Filter Bank

The spectral averaging filter bank should have flat frequency response, thus the triangular windows overlap each other in 50%. The spectral averaging returns the acoustic energy distribution in bands limited by the averaging windows and significantly reduces the speech signal window size for further processing. With the reduction of the number of averaging windows the number of input data for computationally expensive logarithm is reduced, as well as the number of multiplication during DCT computation. The main computational problem is windows overlapping which does not allow to perform straight multiply accumulate operation. The LFCC because of its regularity allows to significantly simplify computation process. The linear filter bank, in comparison to MFCC, requires also less memory for storing window coefficients, since it has linear distribution in frequency domain and every filter has the same coefficient.

It is possible to find such set of averaging windows linearly distributed in frequency domain, that the coefficients of filters are linear combination of powers of 2. That allows to eliminate computational expensive operation of multiplication and replace it by bit shifts and additions only.

The example of such filter bank is presented on the Fig. 2.

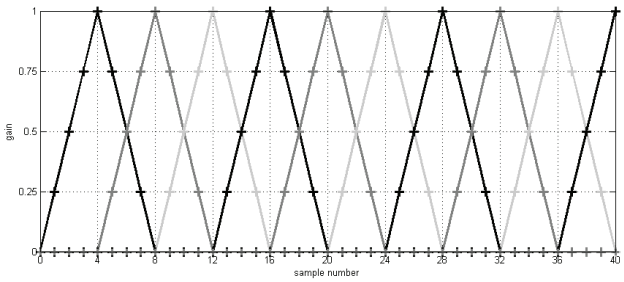

Figure 2. First 9 of the bank of 31 cepstral filters

```
INPUTS:
    npts – number of points of speech windows
    ftrs – number of filters in desired filter bank
RESULT:
    win –the filter bank matrix

//vector of window samples
for i from 0 to npts-1 do
    f[i] = i
end for;

//vector of base points of the filters
for i from 0 to ftrs+2 do
    mf[i] = i*npts/(ftrs+1);
end for;

for i from 0 to ftrs-1 do
    //searching for index in linear freq. vector which freq.
    //corresponds to base point of the window
    for j from 0 to npts-1 do
        if mf[i] == f[j] then f1 = j;
        if mf[i+1] == f[j] then f2 = j;
        if mf[i+2] == f[j] then f3 = j; break;
    end for;

    //finding the  slope and the intercept for lines of filter
    au = 1/(mf[i+1]-mf[i]);
    bu = mf[i]/(mf[i] – mf[i+1]);
    ad = 1/(mf[i+1]-mf[i+2]);
    bd = mf[i+2]/(mf[i+2] – mf[i+1]);

    //forming the vector with the window coefficient
    win[i][] = [au* f[f1+1:f2] + bu , ad* f[f2+1:f3-1] + bd];
end for;

return win;
```

Figure 3. Algorithm for computation of filter bank

In [2] it has been shown that error rate of automatic speaker verification algorithm based on LFCC and DTW decreases when the number of filters grows. Thus such filter bank should be chosen that has coefficients being linear combination of powers of 2 and has required trade-off between the performance (computation costs) and quality (the algorithm error rate).

The Fig. 3 presents pseudo code of the algorithm that allows to compute the required set of filters.

IV. Hardware Implementation

Fig. 4 presents the block diagram of LFCC processor where function implemented in hardware components are shown. The processor was designed to be implemented in FPGA structure and work at 50 MHz frequency.

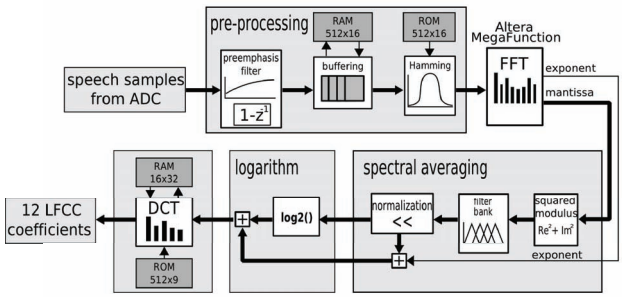

Figure 4. LFCC processor hardware setup scheme

A. Seech Signal Preprocessing

In the preprocessing module speech signal is filtered with pre-emphasis filter and buffered in dual-port RAM memory. It is then split into 256 samples frames with 180 samples overlapping. Next Hamming window function is applied to the signal.

B. Fast Fourier Transform

Fourier transform is implemented as FFT library core. The results are represented in Block Floating Point format, what allows for much better precision than Fixed Point representation. FFT core works in burst mode – all 256 samples of signal window are acquired and then results are computed. FFT core working in this mode requires much less FPGA resources than cores working in other modes.

C. Filter Bank in Spectral Averaging Module

The spectral averaging (filter channels) may be the most computational extensive part of the feature extraction [5]. Using the linear filter bank the computation cost may be significantly reduced up to 45 times. The Fig. 5 presents the concept of hardware implementation of spectral averaging with use of 31 equally distributed windows.

The cepstral averaging is implemented with use of two accumulators, each of them accumulates the values for even and odd windows. The scaling of input samples (multiplication by the value of coefficient) is implemented as combinational circuit performing bit shifts and additions. The scaling of input sample by every coefficient is computed simultaneously. The multiplexers controlled by mod 8 counter are used to selects the appropriate value.

Such cepstral averaging component does not introduce any delay into the circuit. Thank to this it may operate at low frequency equal to the input signal sample rate, what significantly reduces the power consumption, that is proportional to the operational frequency [12].

The complexity of given application depends on the number of spectral averaging filters used. If less filters are used the number of coefficients of each filter increases. Thus the size of multiplexers and the number of combinational circuit for bit shifts and additions increases proportionally to the number of filter coefficients.

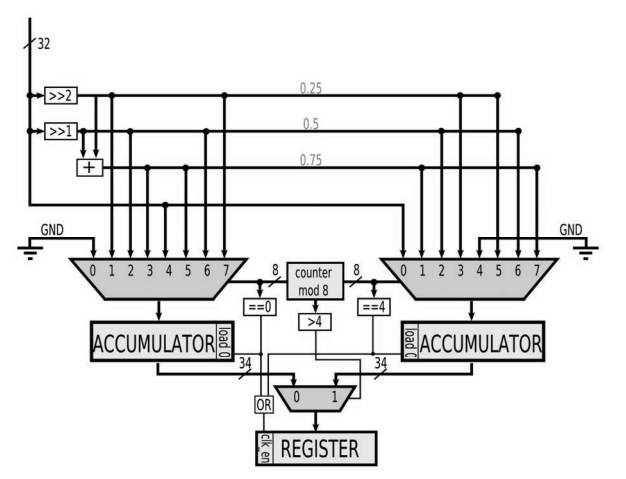

Figure 5. The architecture of spectral averaging circuit with 31 triangular windows, equally distributed in frequency domain

D. Logarithm

The logarithm module is implemented as FSM according to the method described in [13]. It returns 16 bit value in fixed point representation. Computation takes 20 clock cycles. At this moment it is the slowest part of LFCC computation processor and it reduces the overall performance.

E. Discrete Cosine Transform

The values of coefficients necessary for DCT computation are stored in ROM memory. The size of coefficients is selected in a way that ensures most efficient utilization of embedded memory blocks of FPGA structure and can be changed using software application created by the authors.

DCT is computed by using coefficient ROM memory, multiuply-and-accumulate unit, dual-port RAM for storing DCT results and unit that controls access to memory blocks.

The implementation of the LFCC processor was verified with the bit accurate fixed point model presented in [14]. With the dynamic time warping pattern matching procedure and presented LFCC features processor the automatic speaker verification machine has error rate: 1,5%.

V. Results

The LFCC processor described in this paper has been modeled in VHDL and implemented in low cost Cyclone II EP2C35F484C8 FPGA [15]. Quartus II 7.2 CAD tool was used with optimization set to "*balanced*".

The Tab. 1 presents the processing time of single speech frame (256 samples). The "Operation time" presents time necessary to perform given stage of feature extraction algorithm, provided that all necessary data for given stage are available (execution time of stand-alone component). The "Overall time" describes the execution time of given stage with cooperation with other components. It includes also time, when the component waits for data or is back pressured by the subsequent component in the execution flow.

TABLE I. Frame execution time of LFCC processor at 50 MHz

	FPGA at 50 MHz	
	Operation time	***Overall time***
Pre-processing	5.12 µs	5.12 µs
FFT	10.54 µs	23.24 µs
Spectral averaging [a]	2.56 µs	12.00 µs
Logarithm	12.40 µs	12.40 µs
DCT	7.44 µs	12.54 µs
Total	**24.22 µs**	

a. including squared module, filter bank processing and normalization

Thanks to pipelined and parallel structure of proposed LFCC processor the processing of speech frame takes only 24.22 µs and is much smaller than the sum of processing time for each step. The 12 cepstral coefficients are estimated from one 256 sample speech signal frame just in 1,211 clock cycles. The next speech frame is processed when new 180 samples are collected in buffer. The presented solution allows to extract LFCC features at frequency only 10 times greater than speech signal sampling frequency.

Tab. 2 presents the performance comparison of implementation presented in this paper and solution described in [5]. Solution described by Ramos-Lara et al. was implemented on Pentium IV general purpose processor and on a Xilinx Spartan 3XCS2000 FPGA. It can be noticed that presented solution performs very well. It is 10 times faster than both solution designed by Ramos-Lara et al.: FPGA based, as well as Pentium based. The presented solution does not estimate delta coefficients, because they may be estimated offline during the pattern matching. Additionally it was shown in [2] that the dynamic parameters does not introduce significant improvement in quality of speaker recognition (for LFCC speech features and DTW pattern matching) and introduce big computational effort in pattern matching procedure, since with dynamic parameters the feature vector grows two (when using velocity parameters, the first derivative of estimated parameters) or three times (when using velocity and acceleration parameters, 1st and 2nd derivative of parameters)

TABLE II. Frame execution time Comparison

	Method presented in [5]		**Our method**
	Pentium IV at 1.5 GHz	***FPGA at 50 MHz***	***FPGA at 50 MHz***
Pre-processing	17.25 µs	55.96 µs	5.12 µs
FFT	63.36 µs	30.22 µs	10.54 µs
Filter Channels	45.45 µs	116.48 µs	2.56 µs[a]
Logarithm	8.41 µs	53.78 µs	12.40 µs
DCT	102.57 µs	26.46 µs	7.44 µs
Delta coefficients	1.73 µs	2.54 µs	–
Total	**238.77 µs**	**285.44 µs**	**24.22 µs**

a. including squared module, filter bank processing and normalization

The efficiency of described solution allows to estimate speech signal features in real time even if the presented LFCC processor works at frequency just 10 times faster than the speech signal sampling frequency. This is very power efficient solution and it is suitable for battery powered devices like mobile phones netbooks and other personal PDAs.

Tab. 3 presents the resources necessary to implement LFCC processor in Cyclone II EP2C35F484C8. The columns falling under *LE*, *FF*, *RAM*, *MAC* labels describe number of logic elements, flip flops and bits of on-chip RAM memory respectively. Column falling under f_{max} presents the maximal frequency that given component can work with. For comparison solution described in [5] implemented in Spartan 3XCS2000 requires 3817 slices (6138 4 input LUTs) 4659 flip-flops and 13 multipliers.

TABLE III. IMPLEMENTATION RESULTS OF LFCC PROCESSOR IN FPGA

LFCC component	**FPGA resources**				**f_{max}**
	LE	***FF***	***RAM***	***MAC***	**[MHz]**
Pre-processing	85	52	12 288	2	133
FFT	3 428	3 468	14 870	24	105
Spectral averaging [a]	511	151	0	4	86.39
Filter bank	332	107	0	0	111
Logarithm	119	68	0	2	85.30
DCT	127	90	5 120	2	85.07
Total	**4 774 14%** [b]	**3 773 11%** [b]	**32 278 7%** [b]	**34 49%** [b]	**81.25**

a. including squared module, filter bank processing and normalization

b. the part of the overall available FPGA resources

VI. CONCLUSIONS

The presented hardware implementation of LFCC processor is very efficient. It allows to estimate speech features more the 10 times faster than similar solution presented in [5]. The described architecture for spectral averaging optimized for implementation in FPGA architecture significantly improves the overall performance. It solves the most computational extensive part of cepstral feature extraction. The presented solution allows to estimate speech signal features in real time even if the presented LFCC processor works at frequency just 10 times faster than the speech signal sampling frequency, what makes it very power efficient solution suitable for battery powered devices like mobile phones netbooks and other personal PDAs that require voice security facilities. The presented feature extraction module occupies less than 20% resources of the low cost Cyclone II EP2C35 FPGA. This allows to integrate it with other components of speaker verification tool as the System on Programmable Chip and easily fit it into one FPGA.

ACKNOWLEDGMENT

This work was partly supported by the Ministry of Science and Higher Education of Poland - research grant no. N N516 418538 for 2010-2012.

REFERENCES

[1] D. A. Reynolds. Experimental Evaluation of Features for Robust Speaker Identification. IEEE Trans. Speech and Audio Processing, 2(4):639–643, 1994

[2] A.Kaczmarek, M.Staworko, Application of Dynamic Timer Warping and Cepstrograms to Text-Dependent Speaker Verification, Signal Processing Algorithms, Architectures, Arrangements and Applications, 24-26 September, Poznan, Poland

[3] Charbuillet, C., Gas, B., Chetouani, M., Zarader J.L.: Optimizing feature complementarity by evolution strategy: Application to automatic speaker verification. Speech Communication, Vol. 51, No. 9, September, 2009, pp. 724-731.

[4] Choi, W-Y., Ahn, D., Burn Pan, S., Chung, K., Chung, Y., Chung, S-H. "SVM-Based Speaker Verification System for Match-on-Card and its Hardware Implementation", ETRI Journal, vol. 28, no. 3, pp. 320-328, June 2006.

[5] R. Ramos-Lara, M. Lopez-Garcia, E. Canto-Navarro, L. Puente-Rodriguez. SVM speaker verification system based on a low-cost FPGA. Field Programmable Logic and Applications, 2009. FPL 2009. International Conference on (29 September 2009), pp. 582-586.

[6] Nedevschi, S., Patra, R., Brewer, E., "Hardware Speech Recognition for User Interfaces in Low cost, Low Power Devices," 43nd Design Automation Conference, IEEE Press, California, June 2005, pp.684-689

[7] Melnikoff, S., Quigley, S.F., Rusell, M. J., "Implementing a Simple Continuous Speech Recogniton System on an FPGA," Proceedings of the 10th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, Napa, California, USA (2002)..

[8] J.R Deller , Discrete- time processing of speech signal, Macmillan Publ. Co. 1993, New York

[9] Campbell J. P.: Speaker Recognition: A Tutorial, Proc. IEEE, Vol. 85, No. 9, 1997, pp. 1437-1462

[10] Zieliński T.: Cyfrowe przetwarzanie sygnałów: od teorii do zastosowań, Warszawa, WKiŁ, 2005 (in Polish).

[11] S. Hayakawa, F. Itakura. Text-dependent speaker recognition using the information in the higher frequency band, ICASSP, vol. 1, Acoustics, Speech, and Signal Processing, 1994, pp.137-140.Hayakava i Itakura

[12] J. Hannessy, D. Patterson, Computer Architecutre A Quantitive Approach, Fourth Edition, Morgan-Kaufman, 2007

[13] J. C. Majithia, D. Levan, A note on base-2 logarithm computations, Proc. IEEE, pp. 1519-1520, 1973

[14] M. Staworko, M. Rawski, A Data-accurate Model of Automatic Speaker Verification Algorithm for Implementation as System-on-Programmable-Chip in FPGA Structure, II International Interdisciplinary Technical Conference of Young Scientists, 20-22 May, Poznan Poland

[15] Altera Corporation, Cyclone II Device Handbook, Volume 1. 2009, http://www.altera.com/literature/hb/cyc2/cyc2_cii5v1.pdf