IMPLEMENTATION AND ANALYSIS OF GMM-BASED SPEAKER IDENTIFICATION ON FPGA

by

PHAKLEN AL EHKAN
(1040210486)

A thesis submitted
in fulfilment of the requirements for the degree of
Doctor of Philosophy

School of Computer and Communication Engineering
UNIVERSITI MALAYSIA PERLIS

2012

IMPLEMENTATION AND ANALYSIS OF GMM-BASED SPEAKER IDENTIFICATION ON FPGA

PHAKLEN AL EHKAN

UNIVERSITI MALAYSIA PERLIS
2012

UNIVERSITI MALAYSIA PERLIS

DECLARATION OF THESIS

Author's full name : PHAKLEN AL EHKAN

Date of Birth : 23-09-1969

Title : IMPLEMENTATION AND ANALYSIS OF GMM-BASED SPEAKER IDENTIFICATION ON FPGA

Academic Session : Semester II – 2011/2012

I hereby declare that the thesis becomes the property of Universiti Malaysia Perlis (UniMAP) and to be placed at the library of UniMAP. This thesis is classified as:

□ **CONFIDENTIAL** (Contains confidential information under the Official Secret Act 1972)*

□ **RESTRICED** (Contains restricted information as specified by the organization where research was done)*

☑ **OPEN ACCESS** I agree that the thesis is to be made immediately available as hard copy or on-line open access (full text)

I, the author, give permission to the UniMAP to produce this thesis in whole or in part for the purpose of research or academic exchange only (except during a period of _____ years, of so requested above).

Certified by:

______________________________ ______________________________

SIGNATURE **SIGNATURE OF SUPERVISOR**

______________________________ ______________________________

NEW IC NO. / PASSPORT NO.) **NAME OF SUPERVISOR**

Date: ______________________ Date:________________________

NOTES: * If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentially or restriction.

ACKNOWLEDGEMENT

Firstly, I wish to thank The Vice Chancellor, Brigadier General Dato' Professor Dr. Kamarudin Bin Hussin for his constant encouragement and facilities provided at the Universiti Malaysia Perlis for the completion of this research.

I am greatly indebted to my main supervisor, Dean of the School of Computer and Communication Engineering, Universiti Malaysia Perlis, Professor Dr. R. Badlishah Ahmad for his valuable guidance, inspiring advice and continue encouragement as well as support at all stages of this thesis work.

I am also grateful to Dr. Steven F. Quigley and Mr. Timothy Allen from the University of Birmingham for their support, timely suggestion and facilities provided for the completion of this work. I thank them for constantly encouraging me to complete this work.

I wish to thank the Malaysia Ministry of Higher Education for providing the research grants, FRGS9003-00279 under the Fundamental Research Grant Scheme, which funded research work.

Last but not least, I owe my deepest gratitude to my beloved family; parent, wife - Nui Din Keraf, children - Sarayuth Prommanop, Saranyaa Prommanop and Sara Suphamaard Prommanop who have been supported and always behind me throughout my study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADC	Analogue-to-Digital Converter
ANN	Artificial Neural Network
ALU	Arithmetic Logic Unit
ASIC	Application Specific Integrated Circuit
ASM	Algorithmic State Machines
BRAM	Block RAM
CMN	Cepstral Mean Normalisation
CPU	Central Processing Unit
DCT	Discrete Cosine Transform
DSP	Digital Signal Processing
DTW	Dynamic Time Warping
EM	Expectation Maximization
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
FSM	Finite State Machines
GMM	Gaussian Mixture Model
HDL	Hardware Description Language
HMM	Hidden Markov Model
LBG	Linda, Buzo and Gray
IC	Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers
LSB	Least Significant Bit
LPC	Linear Predictive Coefficient
LUT	Look Up Table
MFCC	Mel Frequency Cepstral Coefficient
ML	Maximum Likelihood

MLP	Multi-Layer Perceptron
MSB	Most Significant Bit
NIST	National Institute of Standards and Technology
NN	Neural Network
PIN	Personal Identification Number
PLP	Perceptual Linear Prediction
RAM	Random Access Memory
RASTA	RelAtive SpecTrAl
SVM	Support Vector Machine
VHDL	VHSIC-Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
VQ	Vector Quantization

LIST OF SYMBOLS

c_n	Ceptral Coefficients
f_c	Central frequency
f_{c+1}	Upper pass band
f_{c-1}	Lower pass band
f_s	Sampling frequency
i	$i^{th.}$ component in GMM
$L(\lambda)$	Log likelihood of event λ
mfb	Mel filter bank
n	Samples being evaluated
N	Samples number of windows
NF	Number of filter
$p(i \mid x,\lambda)$	Probability of event i given event x and λ
$p(x)$	Probability of event x
$p(x\mid \lambda)$	Probability of event x given event λ
s	Speaker
S	Size of Speaker
S_k	Mel-scaled signal
w	Component weight
X	Series of feature vectors
x	Individual feature vectors
x_{ij}	Feature vectors
Σ	Covariance
μ	Mean
σ	Covariance Diagonal

PERLAKSANAAN DAN ANALISIS PENGENALPASTIAN PENUTUR BERDASARKAN GMM MENGGUNAKAN FPGA

ABSTRAK

Penggunaan satu sistem pengenalpastian yang mempunyai ketepatan sangat tinggi diperlukan dalam masyarakat kini. Sistem sedia ada seperti nombor pin dan kata laluan mudah dilupai atau dipalsukan dan bukan lagi menawarkan tahap keselamatan yang tinggi. Penggunaan ciri-ciri biologi (biometrik) diterima secara meluas sebagai tahap sistem keselamatan yang lebih tinggi. Salah satu biometrik adalah suara manusia dan ianya menerajui dalam tugas pengenalpastian penutur. Pengenalpastian penutur adalah proses untuk menentukan samada penutur wujud di dalam kumpulan yang telah diketahui dan mengenalpasti penutur di dalam kumpulan itu sendiri. Ciri-ciri penutur wujud dalam isyarat suara disebabkan penutur yang berbeza mempunyai saluran vokal resonan yang berbeza. Perbezaan ini boleh diperlakukan dengan mencungkil Koefisien Kepstral Frekuensi-Mel (MFCC) daripada isyarat suara. Proses pemodelan statistik yang dikenali sebagai Model Bercampur Gaussian (GMM) digunakan untuk memodel taburan setiap MFCC penutur dalam ruang akustik multi-dimensi. GMM terlibat dengan dua fasa iaitu latihan dan pengkelasan. Fasa latihan sangat kompleks dan ianya lebih sesuai dilaksanakan dengan menggunakan perisian. Fasa pengkelasan pula lebih sesuai untuk dilaksanakan menggunakan perkakasan dan ini membenarkan pemprosesan aliran suara masa nyata yang banyak bagi saiz populasi yang besar. Beberapa teknik inovasi telah menunjukkan bahawa sistem perkakasan mendapatkan nilai kelajuan yang lebih tinggi berbanding dengan perisian dengan mengekalkan tahap ketepatan sistem itu sendiri. Melalui pendekatan ini, faktor kelajuan sebanyak lapan puluh enam kali ganda di atas perkakasan FPGA berbanding dengan perlaksanaan menggunakan perisian telah dicapai.

IMPLEMENTATION AND ANALYSIS OF GMM- BASED SPEAKER IDENTIFICATION ON FPGA

ABSTRACT

The use of highly accurate identification systems is required in today's society. Existing systems such as pin numbers and passwords can be forgotten or forged easily and they are no longer considered to offer a high level of security. The use of biological features (biometrics) is becoming widely accepted as the next level for security systems. One of the biometric is the human voice and it leads to the task of speaker identification. Speaker identification is the process of determining whether a speaker exists in a group of known speakers and identifying the speaker within the group. Speaker specific characteristics exist in speech signals due to different speakers having different resonances of the vocal tract. These differences can be exploited by extracting Mel-frequency Cepstral Coefficients (MFCCs) from the speech signal. A statistical modelling process known as Gaussian Mixture Model (GMM) is used to model the distribution of each speaker's MFCCs in a multi-dimensional acoustic space. GMM involves with two phases called training and classification. The training phase is complex and is better suited for implementation in software. The classification phase is well suited for implementation in hardware and this allows for real time processing of multiple voice streams on large population sizes. Several innovative techniques are demonstrated which enable hardware system to obtain two orders of magnitude speed up over software while maintaining comparable levels of accuracy. A speedup factor of eighty six is achieved on hardware-based FPGA compared to a software implementation on a standard PC for this approach.

CHAPTER 1

INTRODUCTION

Speaker recognition, also known as voice recognition is the task of recognizing people from their voice signals (Doddington, 1985). It has a history dating back some few decades where the output of several analogue filters was averaged over time for matching. Speaker recognition uses the acoustic features of speech that have been found to differ between individuals. These acoustic patterns reflect both anatomy (size and shape of the throat and mouth) and learned behavioural patterns such as voice pitch and speaking style. This incorporation of learned patterns into the voice templates has earned speaker recognition its classification as a "behavioural biometric" (Furui, 1994).

The evolution of speaker recognition is quantum jump in artificial intelligence and technology of forensic science because it endows machines with the human-like abilities to distinguish people's identity from one another (Judith, 2000). Speaker recognition technologies are currently applying in many daily applications ranging from police work to automation of call centres. These include the access control system, security control for confidential information, transaction authentication as well as the telephone banking.

The success of speaker recognition system depends largely on how to classify a set of feature used to characterize speaker specific information (Jiuqing and Qixiu, 2003; Sorensen and Savic, 1994). However, pattern classification from speech signal remains as a challenging problem encountered in general speaker recognition system, including speaker verification and speaker identification. Recent development in classifying speaker data from a group of speakers is still insufficient to provide a satisfying result in achieving high performance pattern classification. There are two main difficulties in pattern classification field; first, how to maintain accuracy under incremental amounts of training data and second, how to reduce the processing time as real time systems regarding efficiency and simplicity of calculation (He and Zhao, 2003; Campbell, 2002).

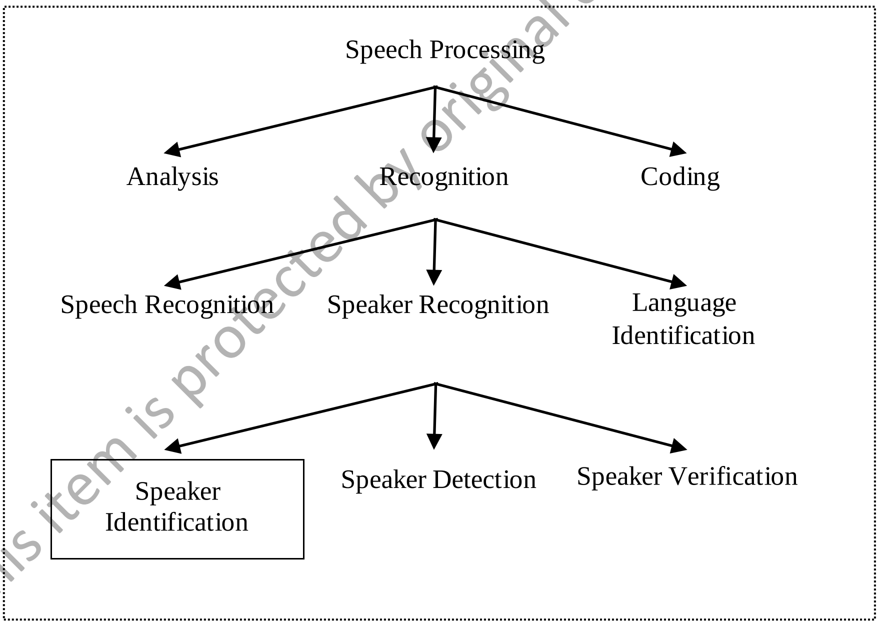


Figure 1.0
Speech Processing Branches (Campbell, 1997)

Figure 1.0 shows the relationship between speech processing and speaker identification branch. Speaker identification is among the most popular method for biometric techniques rely on some physical features that can be unique attributed to an individual besides the iris scanning, face recognition, and digital fingerprint identification. Although the iris scanning and digital fingerprint identifications are extremely accurate indicators of the identity of individual compared to the speaker identification but it is an upcoming and promising technique. Speaker identification systems are popular in spite of their poorer accuracy vis-à-vis the other techniques mentioned earlier because they are the least expensive to build as well as non-invasive in nature (Reynolds, 1995).

In this project, the development in classifying speaker data from a group of speakers is performed on hardware using RC2000 FPGA platform. A satisfying analysis result of the hardware versus software comparison has demonstrated that speaker identification classification is eighty six times faster in hardware. The developed system is capable of processing eighty six times more audio streams in real time than could be done by desktop computer.

1.1 Research Background

The building of robust speaker recognition system is always difficult because of the dynamic speech signal and influences from many sources of variation. There have seen significant progress being made to deal with this problem using different techniques in the past two decades (Sadaoki, 1997). The problem of speaker recognition belongs to a much broader topic in scientific and engineering so called pattern classification. The goal of pattern classification is to classify objects of interest into a number of categories or classes (Richard, Peter, and David, 2000). The categories or classes here are referred to the individual speakers.

The pattern classification plays as an essential part in speaker modelling component chain. The results of it strongly affect the speaker recognition engine to decide whether to accept or reject a speaker. Early pattern classification was produced by Sakoe and Chiba (1978) and Jingwei et al. (2002) through DTW technique and Lawrence (1989) of HMM technique. These techniques are not really efficient for real time application due to characteristic of text dependent recognition. VQ (Vlasta and Zdenek, 1999), GMM and SVM (Solera et al., 2007) as the alternative methods were introduced for speaker recognition to solve the problem. Besides, the GMM classification is the focus of research after Reynolds and Rose (1995) demonstrated its effective performances in text independent speaker identification. The GMM technique of pattern classification in previous studies appeared to have several advantages. However, the process practically does not always produce satisfied result due to the long computational time (Hong et al., 2004; Reynolds and Campbell, 2007). Consequently, alternative methods must be sought in order to reduce processing time problem for GMM technique.

There are some hybrid methods for speaker pattern classification. They draw the attention of the researchers because it was proved with significant improvement for speaker recognition accuracy rates such as hybrid GMM/ANN (Xiang and Berger, 2003), hybrid GMM/VQ (Pelecanos et al., 2000) and hybrid GMM/SVM (Fine et al., 2001; Minghui et al., 2006). Fenglei and Bingxi (2003) claimed that most of these hybrid systems use GMM because it was able be performed in a completely text independent situation. Performance of speaker recognition systems in term of accuracy rate has been significantly improved over hybrid conditions. However, Moon et al., (2003) declared that when speaker recognition is adopted in real-world application, processing time issue is often observed. Meanwhile, current works for the hybrid production of speaker recognition are directed more towards accuracy problems, not processing time problems. Therefore, it is encouraging if a speaker recognition task can be conducted in a "good and fast" pattern classification machine such as in FPGA-based hardware implementation.

To date, most attempts to apply FPGA processing to speech problems focused on the problem of speech recognition (Melnikoff et al., 2002; Miura et al., 2008; Yoshizawa et al., 2006; Lin and Rutenbar, 2009) in which an acoustic speech signal was converted to a text representation of what the speaker has said. Some researchers have been motivated by the desire to achieve a large speedup over real time in order to accelerate searches of multimedia databases. For example, Lin and Rutenbar (2009) demonstrated a 17 times speedup over real time whilst maintaining good recognition accuracy. Other researchers aimed to achieve real-time recognition performance comparable to that of a standard microprocessor, but at much lower power dissipation. For example, Yoshizawa et al. (2006) demonstrated a 10 times improvement in total energy dissipation over a system based on a TMS320VC5416 DSP for real time recognition tasks. Relatively few researchers have investigated the problem of hardware implementation of speaker identification, and these do not aimed to achieve large speedups of performance, but instead to achieve identification using hardware at lower cost than a standard computer system. The speaker identification hardware of (Ramos-Lara et al., 2009) achieved performance comparable to that of a Pentium IV computer for a single voice stream, but using only 24% of the resources of a low cost Xilinx Spartan 3 2000 FPGA.

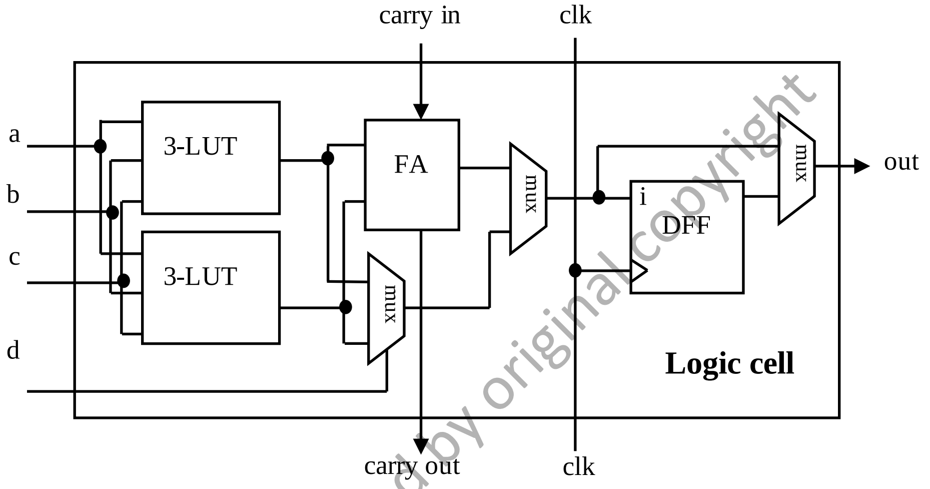
The hardware implementations initially tended to be based on parallel arrays of one kind or another, often using customize chips. As the technology improved, the focus has shifted towards serial implementations, making use once again of customize chips such as application specific integrated circuits (ASICs), microcontrollers or DSPs. Since the appearances of FPGA, that too has been used as a platform of experimental. ASICs customized for a particular use are very expensive even though they provide the highest performance. DSP-based designs, on the other hand, are cost efficient and low in power consumption and heat-emission. However, they only provide a limited speed for data processing because using special memory architectures that are able to fetch multiple data and/or instructions at the same time, they are susceptible to arithmetic saturation. FPGAs are usually slower than ASICs but have the advantage of shorter time to market, ability to be re-programmed in the field for errors correction and upgrades, flexibility, and reducing-cost. Therefore, they combine many advantages of ASICs and DSPs. The use of hardware description languages (HDLs) allows FPGAs to be more suitable for different types of designs where errors and components failures can be limited. Due to the exponential increase of technologies, designers are faced with problems that require the advent of systems that can be fast, flexible, and mainly re-programmable. FPGAs, because of their advantage of real-time in-circuit reconfigurability, make the FPGA based system flexible, programmable, and reliable. They also facilitate the prototyping of complex electronic logic designs.

Recent FPGA shave a very high logic capacity and contain embedded Arithmetic Logic Units (ALUs) to optimize signal processing performance (Brown and Rose, 1996 and Battle et al., 2002). The newest generations of design tools offer libraries of common DSP functions, enabling developers to implement complex systems within a reasonable space of time. FPGAs have been used in many areas to accelerate algorithms that can make use of massive parallelism and improving flexibility. FPGAs are able to exploit pipelining and parallelism in a much more thorough way that can be done with parallel computers using general-purpose microprocessors or a single standard processor (Maslennikov, 2006; Sumedh and Bhoyar, 2012).

1.2 Field Programmable Gate Array (FPGA)

FPGA is a type of semiconductor device that contain programmable logic and interconnections which mostly used in logic or digital electronic circuits. The programmable logic components or logic blocks as they are known may consist of anything from logic gates, through to memory elements or blocks of memories, or almost any element. FPGA supports thousands of gates and popular for prototyping integrated circuit (IC) designs. Once a design is set, hardwired chips will be produced to faster performance. FPGA chip is programmable and reprogrammable which is considered as an advantage of it. In this way, it becomes a large logic circuit that can be configured according to a design, but if changes are required it can be reprogrammed with an update. Thus, if circuit board is manufactured and contains an FPGA as part of the circuit, then this is programmed during the manufacturing process, but can be reprogrammed to reflect any changes. The user programmability gives the user access to complex ICs without the high engineering costs associated with ASICs.

FPGA contains many identical logic cells that can be viewed as standard components. Each design is implemented by specifying the simple logic function for each cell and selectivity closing the switches in the interconnect matrix. The array logic cells and interconnects form a basic building blocks for logic circuits. Complex designs are created by combining these basic blocks to create the desired circuit. The logic cell architecture varies between different device families.



Note: LUT – Look-up Table FA – Full Adder DFF – D flip-flop

Figure 1.1
Illustration of a Logic Cell (FPGA, 2011)

Figure 1.1 shows a simplified illustration of a logic cell. Each logic cell combines few binary inputs to one or two outputs according to a Boolean logic function specified in the user program. In most families, the user also has the option of registering the combinatorial output of the cell, so that clocked logic can be easily implemented. The cells combinatorial may be physically implemented as a small look-up-table (LUT) memory or as a set of multiplexers and gates. LUT devices tend to be a bit more flexible and provide more input cell than multiplexer cells at the expense of propagation delay.