Final Project Report

Table of Contents

Project Name : SOPC-based Voiceprint Identification System		
Team Name : Panda		
Team ID : IN00000004		
Tem Members :	1.	Huan Fang
	2.	Xin Liu
Email Address :	1.	huanf@kth.se
	2.	xinliu@kth.se
Contact No :	1.	0762321822
	2.	0707574312
Instructor :		Prof. Ingo Sander

Design Introduction

As globalization, networking, information and digital era's coming, the demand of high reliability of our identity verification is growing.An efficient mean to this is by authenticating users through biometric methods. Among the existing biometric methods, voice biometrics can be an affordable and accurate authentication technology that has been already successfully and widely employed. Voiceprint, as a basic human physiological characteristics, possess a unique role which is difficult to counterfeit, imitate and replace.As a non-contact identification technology, Voice Recognition Technology is being accepted by the users.

Voice authentication refers to the process of accepting or rejecting the identity claim of a speaker on the basis of individual information present in the speech waveform . It has received increasing attention over the past two decades, as a convenient, user-friendly way of replacing (or supplementing) standard password-type matching.

The authentication procedure requests from the user to pronounce a random sequence of digits. After capturing speech and extracting voice features, individual voice characteritics are generated by registration algorithm. The central process unit decides whether the received features match the stored voiceprint of the customer who claims to be, and accordingly grants authentication.

In this work, the architecture of an sopc-based voiceprint identification system is presented.

1. Voice Recognition Technology Principle

Voice Recognition, also known as the Speaker Recognition, has two categories: speaker identification and speaker verification. Speaker identification is used to determine which one of the people speaks, i.e. "one out of more election" ; and speaker verification is used to determine whether a person specified speaks, i.e. "one-on-one recognition".

According to the voice of different materials, voice recognition can be divided into the text-dependent, and text-independent technology. The text-dependent voice recognition system requires speaker pronounce in accordance with the contents of the text. Each person's individual sound profile model is established accurately. People must also be identified by the contents of the text during recognition to achieve better effect. Text-independent recognition system does not require fixed contents of words, which is relatively difficult to model, but is convenient for user and can be applied to a wide range.

Voiceprint recognition is an application based on physiological and behavioral characteristics of the speaker's voice and linguistic patterns. Different from speech recognition, voiceprint recognition is regardless of contents of speech.Rather, the unique features of voice are analyzed to identify the speaker. With voice samples, the unique features will be extracted and converted to digital symbols, and then these symbols are stored as that person's character template. This template is stored in a computer database, a smart card or bar-coded cards. User authentication is processed inside the recognition system to identify matching or not. The system architecture

block diagram is shown in Figure 1.

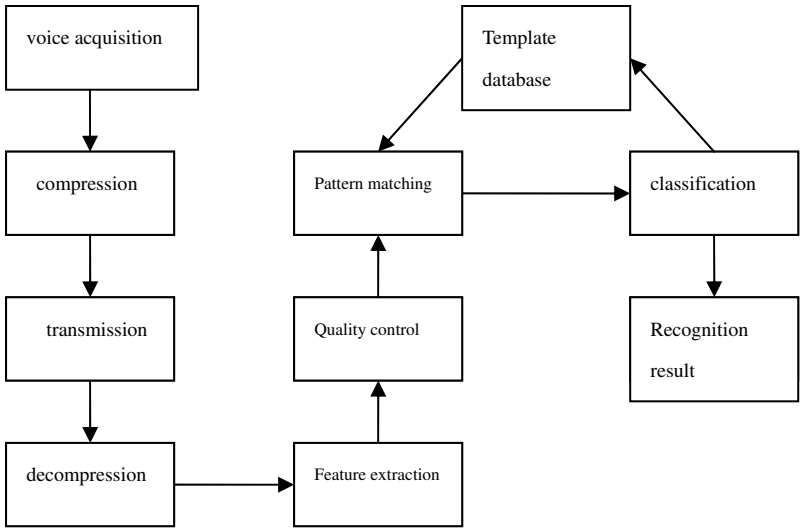


Figure 1 voiceprint recognition system architecture block diagram

2. Hardware Implementation

The Altera DE1 development board features a state-of-the-art Cyclone® II 2C20 FPGA in a 484-pin package. All important components on the board are connected to pins of this chip, allowing the user to control all aspects of the board's operation.
This design is implemented by a 32bit NiosII softcore processor.All IPs are connected on the avalon bus in SOPC builder, including custom peripherals

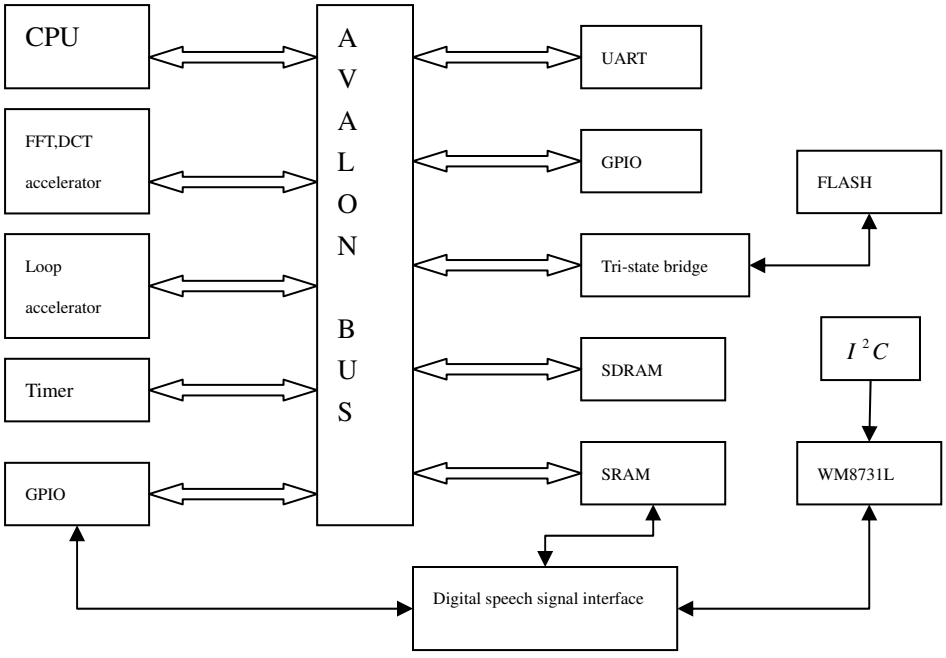


Figure 2 hardware architecture

The system hardware architecture is shown in figure 2,including CPU,uart,tri-state bridge,ram and I/O controls,which are all reusable.Such a design method not only

make it modulization,but also greatly reduce the design cycle of the system.FFT module can not only access IP directly, but also use C2H accelerator tool to improve system performance.In this design ,performance-critical sections such as FFT,DCT and iterative computations will be implemented via C2H hardware accelerator.

Nios II softcore processor

Nios II is a high performance 32-bit sofcore processor. The processor is configured on an Altera Cyclone II FPGA. Custom instructions are added to improve system performance, furthermore, more on-chip rams can be added to improve data processing capacity.

Voice acquisition and verification report

The DE1 board provides high-quality 24-bit audio via the Wolfson WM8731 audio CODEC(enCOder/DECoder). This chip supports microphone-in, line-in, and line-out ports, with a sample rate adjustable from 8 kHz to 96 kHz. The WM8731 is controlled by a serial I2C bus interface, which is connected to pins on the Cyclone II FPGA. A schematic diagram of the audio circuitry is shown in Figure 3.

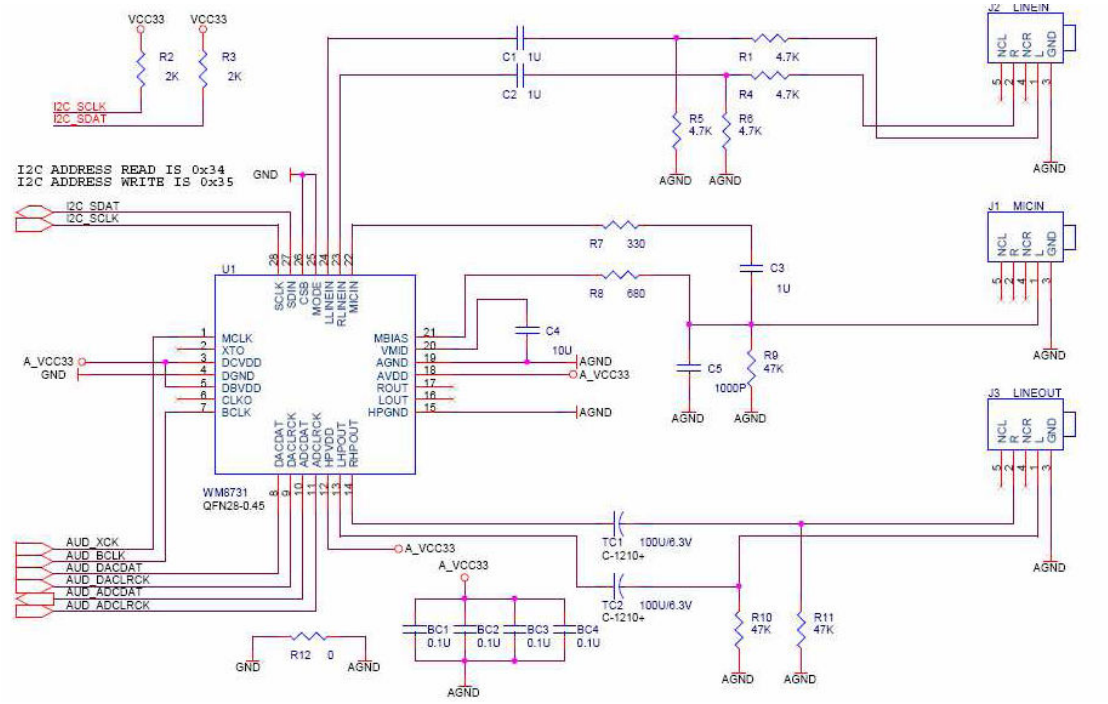


Figure 3 audio schematic diagram

WM8731 contains A/D,D/A modules with a high sample rate and quantization precision.We will use 8kHz sample rate and 16bit quantization precision in this design.

In voice acquisition part, since A / D is the serial data output, a serial to parallel data conversion and control of the SRAM Verilog module is needed.Voice report is communicated with CPU via GPIO,different voice is played according to different verification result.GPIO control is done in Nios IDE. Similarly, since the voice broadcast from FLASH are read out in parallel, thus a parallel to serial data conversion verilog module is needed.

C2H Hardware Acceleration

The Nios® II C-to-Hardware Acceleration (C2H) Compiler is a tool that allows you to create custom hardware accelerators directly from ANSI C source code. A hardware accelerator is a block of logic that implements a C function in hardware, which often improves the execution performance by an order of magnitude. Using the C2H Compiler, you can develop and debug an algorithm in C targeting an Altera® Nios II processor, and then quickly convert the C code to a hardware accelerator implemented in a field programmable gate array (FPGA).

The C2H Compiler improves the performance of Nios II programs by implementing specific C functions as hardware accelerators. The C2H Compiler is not a tool for creating arbitrary hardware systems using C as a design language. What the C2H Compiler does do is to generate an accelerator which is functionally identical to the original C function.

Based on these premises, the C2H Compiler's design methodology provides the following features:

■ ANSI C compliance – The C2H Compiler operates on plain ANSI C code, and supports most C constructs, including pointers, arrays, structures, global and local variables, loops, and subfunction calls. The C2H Compiler does not require special syntax or library functions to specify the structure of the hardware. Unsupported ANSI C constructs are documented.

■ Straightforward C-to-hardware mapping – The C2H Compiler maps each element of C syntax to a defined hardware structure, giving you control over the structure of your hardware accelerator.

■ Integration with C language development environments for the Nios II processor, including the Nios II integrated development environment (IDE), and the BSP generator and related tools. You control the C2H Compiler with the Nios II C development tools. You do not need to learn a new environment to use the C2H Compiler.

■ Based on SOPC Builder and Avalon system interconnect fabric – The C2H Compiler uses SOPC Builder as the infrastructure to connect hardware accelerators into Nios II systems. A C2H accelerator becomes a component within an existing Nios II system. SOPC Builder automatically generates system interconnect fabric to connect the accelerator to the system, saving you the time of manually integrating the hardware accelerator.

■ Reporting of generated results – The C2H Compiler produces a detailed report of hardware structure, resource usage, and throughput.

3. Software Algorithm

MFCC is currently the most popular feature coefficient used in the speech recognition, and it can obtain the more accurate results of speech recognition under a non-noise condition. In the MFCC algorithm, we first use the FFT to calculate the signal frequency spectrum , then we use DCT to further reduce the speech signal’s redundant information, and reach the aim of regulating the speech signal into feature coefficients

with small dimensions. The FFT and DCT algorithm can be used for any speech segment whose time-frequency resolution is fixed.

Feature extraction

MFCC feature coefficient extraction flow chart is shown as Fig. 1. The working process is:

1. Pre-emphasis of the speech signal, frame, adding window, then make the FFT to obtain the frequency information.
2. Pass the signal through the Mel frequency coordinate triangle filter groups to mimic the human hearing mechanism and the human hearing sensibility to different speech spectrum.
3. Calculate the logarithm value of the signal after the Mel filters to obtain the logarithmic spectrum.
4. Make the discrete cosine transform to the signal and obtain the MFCC coefficients.

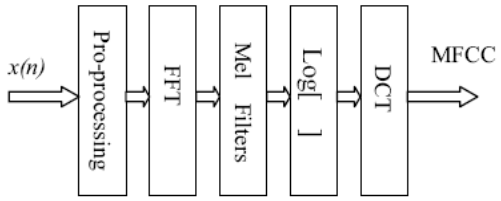
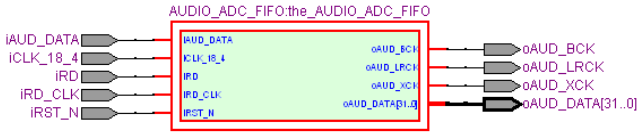


Fig.1 The feature extraction of the
Mel frequency cepstrum coefficients

4. Design Architecture

As seen in SOPC builder, we add an AUDIO_ADC_FIFO_0 module



to convert serial data to parallel audio samples(16 bit). On-chip rams (BufferRAM, CosRAM, SinRAM) are used to store data needed by FFT module which is C2H accelerated.

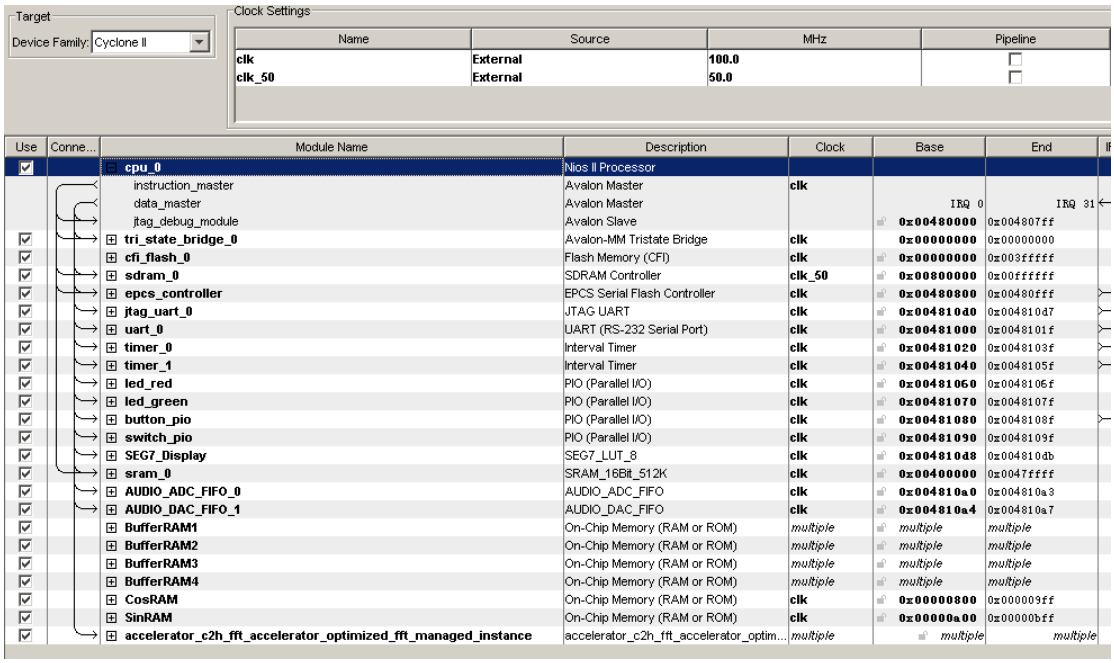


Figure 4 . SOPC builder system

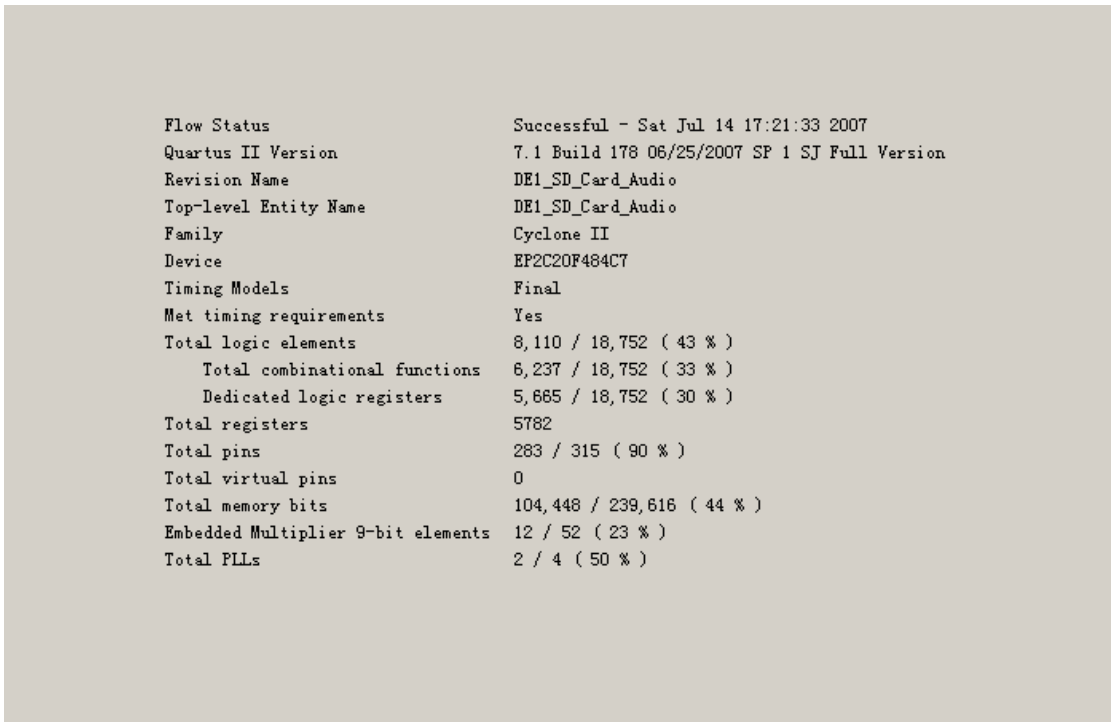


Figure 5 . Final Compilation report

Speech Acquisition

Speech acquisition requires a microphone coupled with an amplified ADC to receive the voice speech signal, sample it, and convert it into digital speech for input to the FPGA. The DE1 development board has a WM8731 audio codec chip connected both to the microphone input pins and the Altera Cyclone II FPGA pins through an I2C serial controller interface. The WM8731 is a versatile audio codec that provides up to 24-bit encoding/decoding of audio signals in various sampling rates ranging from 8 to

96 KHz. The codec clock input is generated by dividing the system clock by four using a custom hardware block. The block combines the clock divider logic and the I2S digital audio interface logic as well as options for programming the control registers. The DE1 board's microphone input port connects the microphone and headset for speech acquisition. The following table shows the codec's control register settings:

Codec Register Settings

Register	Setting
ADCDAT	16 bit
USB/normal mode	Normal mode
Master/slave mode	Slave
Digital audio interface	I2S interface
MCLK input	50 MHz divided by 4
ADC sampling rate	8 KHz

These settings are programmed by setting or resetting of various bits in the control registers as shown in the following table:

Register	Address	Register Name	Value
R0	0000000	Left line in	001A
R1	0000001	Right line in	021A
R2	0000010	Left headphone Out	047B
R3	0000011	Right headphone Out	067B
R4	0000100	Analog audio path control	0815
R6	0000110	Power down control	0C00
R7	0000111	Digital audio interface format	0A06
R8	0001000	Sampling control	100E
R9	0001001	Active control	1201

Original sound wave and frequency spectrum after FFT in MATLAB:

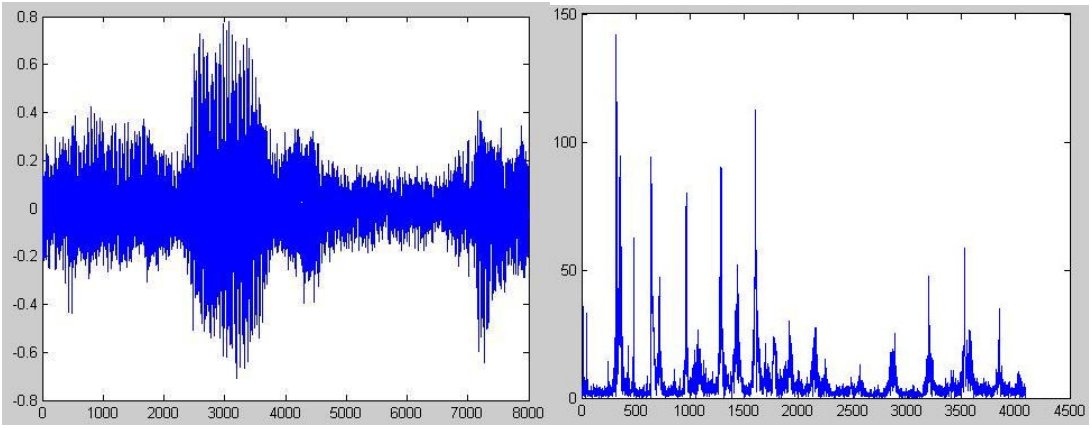


Figure 6: sound data in time domain and frequency domain

Mel Frequency Cepstral Coefficients (MFCCs)

MFCC are coefficients that represent audio. They are derived from a type of cepstral representation of the audio clip (a "spectrum-of-a-spectrum"). The difference between

the cepstrum and the mel-frequency cepstrum is that in the MFC, the frequency bands are positioned logarithmically (on the mel scale) which approximates the human auditory system's response more closely than the linearly-spaced frequency bands obtained directly from the FFT or DCT. This can allow for better processing of data, for example, in audio compression. However, unlike the sonogram, MFCCs lack an outer ear model and, hence, cannot represent perceived loudness accurately.

MFCCs are commonly derived as follows:

1. Take the Fourier transform of (a windowed excerpt of) a signal
2. Map the log amplitudes of the spectrum obtained above onto the mel scale, using triangular overlapping windows.
3. Take the Discrete Cosine Transform of the list of mel log-amplitudes, as if it were a signal.
4. The MFCCs are the amplitudes of the resulting spectrum.

Dynamic Time Warping (DTW)

A distance measurement between time series is needed to determine similarity between time series and for time series classification. Euclidean distance is an efficient distance measurement that can be used. The Euclidian distance between two time series is simply the sum of the squared distances from each nth point in one time series to the nth point in the other. The main disadvantage of using Euclidean distance for time series data is that its results are very unintuitive. If two time series are identical, but one is shifted slightly along the time axis, then Euclidean distance may consider them to be very different from each other. Dynamic time warping (DTW) was introduced to overcome this limitation and give intuitive distance measurements between time series by ignoring both global and local shifts in the time dimension.

Problem Formulation. The dynamic time warping problem is stated as follows: Given two time series X, and Y, of lengths |X| and |Y|,

$$X = x_1, x_2, \ldots, x_i, \ldots, x_{|X|}$$
$$Y = y_1, y_2, \ldots, y_j, \ldots, y_{|Y|}$$

construct a warp path W

$$W = w_1, w_2, \ldots, w_K \qquad \max(|X|, |Y|) \le K < |X| + |Y|$$

where K is the length of the warp path and the kth element of the warp path is

$$w_k = (i, j)$$

where i is an index from time series X, and j is an index from time series Y. The warp path must start at the beginning of each time series at w1 = (1, 1) and finish at the end of both time series at wK= (|X|, |Y|). This ensures that every index of both time series is used in the warp path. There is also a constraint on the warp path that forces i and j to be monotonically increasing in the warp path, which is why the lines representing the warp path in Figure 1 do not overlap. Every index of each time series must be used. Stated more formally:

$$w_k = (i, j), w_{k+1} = (i', j') \qquad i \le i' \le i+1, \;\; j \le j' \le j+1$$

The optimal warp path is the warp path is the minimum-distance warp path, where the distance of a warp path W is

$$Dist(W) = \sum_{k=1}^{k=K} Dist(w_{ki}, w_{kj})$$

Dist(W) is the distance (typically Euclidean distance) of warp path W, and Dist(wki, wkj) is the distance between the two data point indexes (one from X and one from Y) in the kth element of the warp path.

5. Performance Parameters

Recognition Accuracy

We record the word “hello” of user1 for our test voice and store it in the database.

Test case1: user1 speaks “hello” recognition accuracy: 80%

Test case2: user1 speaks other words rejection accuracy: 92%

Recognition Speed

FFT calculation: 100 ms

MFCC feature extraction: 8~12s

DTW algorithm: 1~2s

Total recognition time: 9~14s

6. Conclusion

Our project implements a voiceprint identification system in which a Nios II processor performs the recognition process. We implemented computationally intensive tasks with C2H accelerator and the SOPC Builder helped us integrate these blocks into the Nios II system. However, some tasks can not be C2H accelerated because C2H does not support float and double data type. These processes consumes a large amount of time. Recognition time can be reduced if a faster processor is used.