Abstract—This paper is about a speech recognition system for robot control using the DE2 development kit, which is being used at Computer Engineer Department of the University of Information Technology. Hardware devices of the system are an Altera DE2 development kit and a Philips SHM1000 microphone. The system includes a hardware design in the Verilog HDL and software in Embedded C for Nios II. The core of the system is the hardware on the FPGA, it includes four main components: module for receiving and converting audio signal; memory controller; FFT module and a Nios II processor. The system has 2 modes: Training and Recognition, based on the vector quantization approach.

Index Terms—speech recognition, fast fourier transform, Mel-Frequency filter, audio cepstrum, vector quantization, DE2, nios, verilog, robot control, vietnamese.

I. INTRODUCTION

Speech recognition in human – robot interaction have been investigating and developing by many organizations in the world. Some noticeable achievements are Speech Recognition (Microsoft), HTK (Machine Intelligence Laboratory, Sphinx (CMU)... Most of solutions listed above usually run on high speed computers with large resource requirement. They have not been capable of being integrated in particular purposes system that requires a tiny low power and low resource requirement such as control robots, machines or family devices [1][2].

There are some studies and experiments on speech recognition on FPGA, such as “The Speech Recognition Chip Implementation on FPGA” [3], “An FPGA Implement of Speech Recognition with Weighted Finite State Transducers” [4]. However, in general, those studies just concentrate on recognition but have not been applied to interacting with the robots, and absolutely, have not been designed to work with Vietnamese.

Inspired with that idea, we decide to make a speech recognition system on FPGA for robot control, using the DE2 development kit [7] (available in Computer Engineer Department’s Laboratory – University of Information Technology) for studying/ researching purpose of the department.

The system is based on the hardware design in Verilog HDL using Quartus design tool, and some programmed modules (of Terasic) such as FFT modules, SDRAM controller.

II. SYSTEM SCHEMA

This paper is proposing a system schema included with hardware and software.

Hardware system includes modules as in Fig. 1: ADC, Memory Controller, FFT Controller and Nios II.

Dataflow:
- Speech input (analog signal) from microphone is got into the system after converted to digital signal through the ADC module.
- The digital signal (in time domain) is converted to frequency domain using FFT module.
- The digital signal (in frequency domain) is got into Nios II and to be processed here. The recognition process is carried out by a C program, based on vector quantization approach.
- Output is the command signal (to the robot control) corresponding to input speech.

A C program is embedded in the Nios II processor. It handles the processing and carrying out training/ recognition process, displaying input/ output data. Speech data is converted from time domain to frequency domain here by FFT module, then it is extracted to get speech characteristic. In the training mode, speech characteristic data will be grouped using K-means clustering, this form the codebook – data pattern to be compared with the input speech.

The recognition process is executed by comparing extracted speech characteristic vectors of input speech with the vectors in codebook (trained before by training mode, through the same data flow).

The result is displayed on LCD and indicating LED; the output control signal is transmitted to GPIO pins of...
the DE2 board. Robot receives control signal from here and act as corresponding action (programmed before).

III. SYSTEM DESIGN

A. Hardware

Hardware in DE2 board handles the speech receiving, storing it into memory, segmenting, and Fast Fourier transformation process.

ADC Module: This is start point of the system. This module receives signal from microphone (analog), calculates, samples and returns digital signal with configurable sample rate and resolution. The digital signal data is stored into RAM, waiting for being processed.

Memory Controller Module: This module carries out data storing function (using RAM-onchip memory) after speech signal is converted to digital. The data storage capacity can be changed (depend on control signal).

FFT Controller Module: This gets data from memory, segments the data (with 1/3 overlap ratio) and controls the converting process from time domain to frequency of the FFT module (from Megacore Function Library).

* Data segmenting Fig. 2:
  - Data from memory is segmented into overlapping frames.
  - There are \( N \) samples in a frame. The distance between two frames is \( M \) samples.
  - \( M = \left(\frac{1}{3}\right) N \)

Speech data will be segmented into overlapping frames.

![Data segmenting](image)

**Example:** If each frame has 300 sample, the second frame will begin from sample number 100 (\( M=100 \)).

* FFT controlling diagram Fig. 3
  - **RESET:** default state (initialization, reset). Values are initiated in this state.
  - **INWAIT:** wait for the FFT iStart – start converting signal (from software).
  - **INSOP (Start of Packet):** actives control signals, samples the first data block received from memory.
  - **INMID:** samples all of the remaining data received from memory, until the number of sampled sample (incount) is less than required sample number (LEN-2, with LEN is the FFT size which is 256 in this system).
  - **INEOP (End of Packet):** Get the last sample, and active the stop-receiving-data signal, begin the FFT process.

![FFT Controller States](image)

NIOS II Module: The processor handles the analyzing process, included training and recognition mode, after data is transformed by FFT. Nios II components are described in Fig. 4.

![NIOS II components](image)

- **Nios II Processor:** The type is Nios II/f, 4KB data cache, level 2 and integrated floating point.
- **SDRAM Controller:** communicate with the 8MB SDRAM - main memory of the system
- **Peripheral buses:** iKeys, iSwitchs, oLEDR, oLEDG, LCD, SEG7…
- **iFFTCoeff, FFT_exponent:** FFT module data bus.
- **System control signal:** oStart, iFFCComplete…

The Nios II processor operates at the rate of 100MHz, SDRAM at 100MHz -3ns to get the system to be synchronized.

The training and recognition process is described in the next section.

B. Software

We use a Nios II Application because of its ease of use. Analysis process is programmed in C, and is embedded in Nios II to control the system operation.
The Vector Quantization approach is used in both training and recognition mode.

**Training mode:** Each word is spoken several times, the system will analyze, collect and classify data into a codebook (a vector collection, smaller than initial collection), particularly. The process is described in Fig. 5.

![Diagram](image)

**Figure 5.** The training process

**Speech detection:** At first, the system continuously checks if the input signal is a speech or not by comparing the output exponent from the FFT Controller module (stored in FFT_exponent) with a threshold parameter. The higher frequency of the speech makes the higher exponent (because of shifting process for ensuring the bus data width). After some experiments, we have got that if the FFT_exponent is less than 61, there is a human voice.

**MFCC filter** [5][6]: This is for extracting speech characteristic data according to listen capability of human ear (Mel – Frequency). We designed Mel filter banks as Fig. 6.

**Filter expression:**

\[
H_m(k) = \begin{cases} 
0 & \text{for } f(k) < f_c(m - 1) \\
\frac{f(k) - f_c(m - 1)}{f_c(m) - f_c(m - 1)} & \text{for } f_c(m - 1) \leq f(k) < f_c(m) \\
\frac{f(k) - f_c(m)}{f_c(m) - f_c(m + 1)} & \text{for } f_c(m) \leq f(k) < f_c(m + 1) \\
0 & \text{for } f(k) \geq f_c(m + 1) 
\end{cases}
\]  

(1)

![Graph](image)

**Figure 6.** Mel filter banks in frequency domain

With:
- \(f_c(m)\): center frequency of the \(m^{th}\) filter
- \(F_s\): speech input sample rate
- \(F(k) = kF_s/n\): frequency of the \(k^{th}\) sample of \(n\) samples
- Center frequency \(f_c(m)\) is a linear frequency in Mel field, or the logarithm in normal frequency scale. The conversion expression to Mel scale:

\[
f_{mel} = 2595 \times \log_{10} \left(1 + \frac{f}{700}\right)
\]  

(2)

Computing output energy at each filter: This energy is calculated by log of sum of the products of signal’s frequency amplitude and corresponding weight in the filter.

\[
e_m = \log \sum_{j=1}^{N}(h_m(j) \times X(j))
\]  

(3)

**K-means clustering:** To reduce the training vector collection which make the training codebook. This is based on Euclidean distance formula. Expression for calculating the distance (or space) of two vector:

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

(5)

With:
- \(x, y\): vectors to be compared
- \(P\): vector size (\(P = 12\) in this system)

**Operation steps:**
- **Initialization:** Identify codebook size by randomly choosing \(N\) vector for codebook collection (each codeword is a vector). Corresponding to each codeword is the center of the vector cluster.
- **Find nearest neighbor vector:** for each vector from training collection, calculate the Euclidean distance to get the nearest codeword with the vector, label it (to know that it belongs to that codeword cell).
- **Update the center:** Corresponding to each cell, update the codeword which is the center of all the vectors belonging to that cell.
- **Repeat 2-3, until there is not any vector changes the cells.**

**Recognition mode:** Initial steps are similar with the training mode. After detecting the speech and extracting MFCC characteristics for the received speech, compare the MFCC characteristics vector with each codebook in
the training collection to get out the needed speech. The recognition process is described in Fig. 7:

![Recognition process](image)

**How to get the appropriate codebook for input speech?**

Assume that after the speech characteristics extraction, we have the vector collection \(T(x_1, x_2, x_3, \ldots, x_T)\).

There is \(V\) codebook for vocabulary set and the codebook of \(i^{th}\) word is \(\{y_{1i}, y_{2i}, y_{3i}, \ldots, y_{mi}\}\) with \(m\) is the codebook size.

To get the appropriate codebook for the input speech, we calculate the distance of the characteristics vector with each of codebook in the training collection.

The distance calculating expression with \(i^{th}\) codebook:

\[D_i = \frac{1}{T} \sum_{t=1}^{T} \min_{1 \leq m \leq M} |d(x_t, y_{mi})| \quad (6)\]

From this, we can identify which codebook has the minimum distance \(D_i\) which is greater than recognition threshold with characteristics vector input \(T\), as the following formula:

\[j = \arg\min_{1 \leq i \leq V} [D_i] \quad (7)\]

**IV. RESULT**

The system has been finished included with a hardware design and the associate software (Nios II Application) to control the operation of the hardware, carry out the training and recognition process.


Control signal:

- **Switch SW [0]:** chose the recognition mode.
- **Switch SW [1]:** chose the training mode.
- **Switch SW [2]:** chose the fixed training data from a text file or the new training data on-the-go.

System states and indicators:

- Waiting state: when there is no operation, LCD shows “Waiting…” 7-segment leds display all zero Fig. 8.
- Recognizing/ operating state: **leds LEDR [3]-11** indicate the recognized command to control the robot, the LCD displays “Recognized: <command>”.

The accurate threshold: this is based on the minimum distance \(D_i\) in the recognition mode. According to experiments, \(2.2 < D_i < 3.6\) make the best recognition result. If \(D_i\) is out of this range, the input speech is not on the vocabulary set.

In Fig. 9 and Fig. 10, the system is command the robot to do the “Trái” and “Nhanh” command, **led LEDR [5] and led LEDR [7]** is on corresponding to received command. The 7-segment leds shows some command parameters. Recognized command is display on the LCD.

Some experiment results are showing in the Fig. 8, Fig. 9, Fig. 10.

Because of the accent differences, the system operates more accurately with standard and common voice.

Each command in vocabulary set is spoken 10 times in 4 various voices which is not so different from the trained voice, the result accuracy is averagely greater than 80%.

**TABLE I. EXPERIMENT RESULT**

<table>
<thead>
<tr>
<th>Command</th>
<th>Trái (go)</th>
<th>Lùi (back)</th>
<th>Trái (left)</th>
<th>Phải (right)</th>
<th>Nhanh (fast)</th>
<th>Chậm (slow)</th>
<th>Vừa (medium)</th>
<th>Dừng (stop)</th>
<th>Xoay (rotate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>90</td>
<td>80</td>
<td>80</td>
<td>100</td>
<td>80</td>
<td>80</td>
<td>90</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

Journal of Automation and Control Engineering Vol. 1, No. 3, September 2013
The paper has presented a speech recognition system on FPGA, for a particular purpose – robot control which can work with Vietnamese input speech. We have designed the hardware based on DE2 components and programmed a software and embedded in the Nios II to control the operation of the system.

The recognition method and algorithm is simple, but experimental results have shown that it is completely capable of implicating our system to control other device (here is robot). This lead to the possibility of making integrated chips in small control system, such as controlling the robots, family devices or cars by speech command.

REFERENCES


Phan Dinh Duy was born on October 26, 1988 in Binh Dinh province, Vietnam. He obtained his B.S degree in Computer Engineering from the University of Information Technology where he is working on Circuit Design and machine learning.

Vu Duc Lang received the B.S. and M.S. degree in computer engineering from Saint Petersburg State Polytechnical University in 1998 and 2000, respectively. He got the Ph.D. degree in computer science from Saint Petersburg Electrotechnical University in 2006. From 2006 until now, he works at the University of Information Technology, VNU HCMC as a lecturer. His research interests include machine learning, human-computer interaction and FPGA technology.

He is a member of IEEE, ACOMP 2011 and Publication chair of ICCAIS 2012.