A DIGITAL CHIP FOR ROBUST SPEECH RECOGNITION IN NOISY ENVIRONMENT

Chang-min Kim and Soo-Young Lee

Department of Electrical Engineering
Korea Advanced Institute of Science and Technology

ABSTRACT

A digital chip has been developed for isolated word recognition in real-world noisy environments. By carefully comparing recognition performance and hardware implementability a modified-ZCPA model and RBF neural network model are selected for the feature extractor and classifier, respectively. The modified-ZCPA model is based on feature extraction mechanism of human auditory system, and demonstrated superiority for noisy speeches. The RBF network has excellent OOV (out-of-vocabulary) rejection capability as well as good recognition performance. Both the feature extractor and classifier are implemented by repetition of simple operations, which result in reduction of memory operations for the full use of memory bandwidth. The chips is custom designed at logic level without any DSP core, and implemented at an FPGA with 12 MHz clock speed.

1. INTRODUCTION

Speech recognition is regarded as the major man-machine interface. Although current speech recognition systems require powerful computers with fast processors and large memory, there exist many stand-alone applications in real-world. Speech recognition chips are ideal for these stand-alone applications. However, the performance of speech recognition systems is seriously degraded in the presence of background noise, which becomes more serious in out-door stand-alone applications. Therefore, speech recognition chips require noise-robustness as well as efficient low-power implementation. Careful design study is also essential to make compromise between high performance and efficient VLSI implementation. For the robust feature extraction we adopted the ZCPA (Zero-Crossings with Peak Amplitude) model based on human auditory system, which demonstrated superior performance in noisy environments.[1] Although HMM (hidden Markov model) and neural network models have been commonly used for the classifier, the latter is advantages for VLSI implementations due to its regular computing architecture. Especially, RBF neural networks is chosen for excellent balance between high recognition performance and OOV (out-of-vocabulary) rejection performance. Also, speaker adaptation is easily implemented with the RBF networks. In this paper a new digital chip is reported for isolated word recognition in real-world noisy environments. The feature extraction module and classifier module are described in Section 2 and Section 3, respectively. Section 4 presents the developed chip configuration and test results. Fig. 1 shows the block diagram of the proposed system.

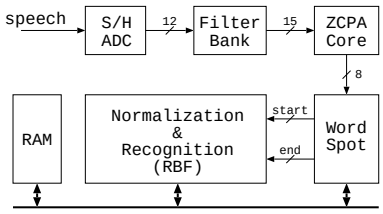


Fig. 1. Block Diagram of the developed speech recognition chip

2. FEATURE EXTRACTION

Fig. 2 represents a block diagram of the ZCPA model[1]. The ZCPA model is composed of cochlear bandpass filters and a nonlinear stage at the output of each bandpass filter. The bank of band pass filters simulates frequency subjectivity of the basilar membrane in the cochlear, and the nonlinear stage models inner hair cells attached to the basilar membrane. The nonlinear stage consists of a zero-crossing detector followed by the compressive nonlinearity. In this paper, the cochlear bandpass filter is constructed by digital FIR filters with powers-of-two coefficients for multiplierless chip implementations. Then, we propose a modified feature extraction method to reduce computational resources.

2.1. Cochlear Filter Banks

Cochlear filter banks are composed of 16 channels and 100 delay taps with quantization at signed 12-bit.

2.1.1. Powers-of-two Decomposition

The FIR coefficients are represented by

$$C_j = \sum_k c_k \times 2^k, c = -1, 0, +1 \qquad (1)$$

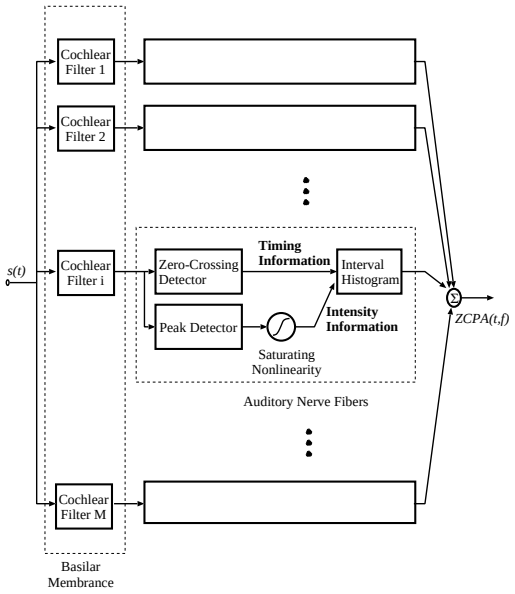


Fig. 2. Block Diagram of the ZCPA model

With this expression, we can design multiplierless FIR filters[5] but the system needs more instructions. By using the symmetry of FIR coefficients, the computing time is reduced by a half. It means that one coefficient has two operands. Now, instructions for the FIR system must be considered. The instructions must include the information of coefficients and operands. The system with multipliers needs one coefficient [12bit] and two operand addresses [6+7=13bit]. Because the filter bank is designed using 100 delay taps at the maximum, the first operand is limited by 50 delay taps. Consequently, it needs 25 $\times$ 324 = 8100 bits overall. Then each powers-of-two filter needs one sign bit, 4 shift bits, and 13 bits for the address. Because the width of FIR coefficients is 11 except the sign bit, the maximum shift is restricted by 4. Accordingly, it needs 17 bits. However, all of powers-of-two coefficients need not to have address information. These must have the address information only if the operands are changed. Because the coefficients are transferred from one multiplier operation to several sign and shift operations, we need 5 $\times$ 723 (sign and shift information) + 13 x 324 (address information) = 7827 bits. If we assume that buffer memory used for the 100 delay taps has a single I/O, the system must have a stall stage for two operands. The multiplier filter needs 648 clocks and the powers-of-two filter needs 723+324=1047 clocks. Naturally, both need clocks for channel information. Therefore, powers-of-two filter need more clocks but less area due to less instruction size and multiplierless operation. In this paper, 12 MHz clock is used to complete the FIR filter operation within time intervals for 11.025KHz sampling.Table 1 shows the comparison between FIR filters with and without a multiplier.

Table 1. The comparison of FIR filters

	powers-of-two	multiplier
non-zero coeff.	722	324
instruction rom size	7827	8100
execution time (# of clocks)	1047	648

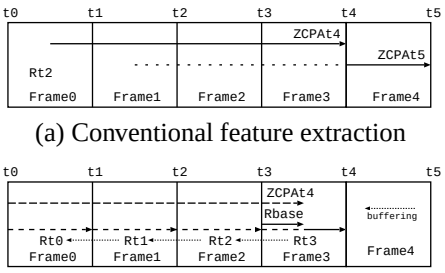
2.2. ZCPA Feature Extraction

The output of ZCPA module at time m is described as

$$y(m,i) = \sum_{n=1}^{N_c h} \sum_{l=1}^{Z_k - 1} \delta_{i j_l} g(A_k l), \quad 1 \le i \le M \quad (2)$$

$$g(x) = \log(1+x) \quad (3)$$

where N is the number of frequency bins, and δ_{ij} is the Kronecker delta. For each channel, the index of frequency bin, j_l, is computed by taking the inverse of the time interval between the lth and the (l+1)th zero-crossings, for l=1, $\cdots$, Z_k-1. Then the value of the frequency histogram at frequency bin, j_l, is increased by $(g(P_k l))$. In the ZCPA model, the length of windows is set to $10/F_k$ where F_k is the center frequency of the kth channel to capture about ten periods of the signal at each channel, provided that the signal is a sinusoid with a single frequency equal to the characteristic frequency of the channel[2]. Thus, the window length becomes longer for low frequencies, and shorter for high frequencies. And the features are generated every 10 milliseconds. Therefore, the conventional ZCPA model needs the memory for speech data as much as the channel of the longest window, which corresponds to the lowest frequency bin.[1]. Fig. 3(a) shows an example of the conventional



(a) Conventional feature extraction

(b) Proposed feature extraction

Fig. 3. Example of Feature Extraction Method

ZCPA model. It needs memories for the previous speech data and checks the zero-crossing at every 10 ms along with the lines in Fig. 3(a). The dotted lines denote the waste of the conventional method. If the system can extract the feature vector during every 10 ms and buffer this information to previous frame, there is no necessity for keeping the speech data and computing include zero-crossing detection

for the previous data. Therefore, we can extract the ZCPA features not from the past speech data but from the buffered feature vectors. For example, we can extract features from Rbase+Rt0+Rt1+Rt2 at t4. The features are extracted by only once computation during 10 ms with only small extra memory due to the buffered feature. For example, the feature of the lowest channel is extracted at 50 ms, equivalent to 550 samples. It needs the memory for 550 filter outputs. But, in the proposed method, only 5 registers are needed, and the ZCPA model works in real time with smaller memories. Fig. 4 shows the ZCPA feature vectors of the proposed system for /i/.

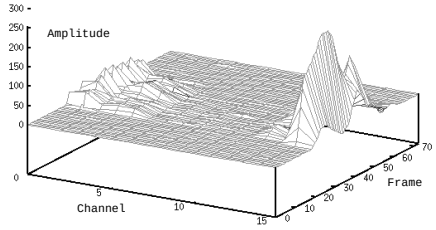


Fig. 4. ZCPA feature vector of '/i/'

3. CLASSIFIER

The classifier based on artificial neural networks can be easily implemented in VLSI due to its regular computing architecture. The operation is composed by the repetition of simple operations, multiplications and accumulations mainly. Especially, Radial Basis Function (RBF) model has the OOV (out-of-vocabulary) rejection capability and realizes the speaker adaptation function by just adding the patterns at a new hidden neuron. Due to these advantages, we design the classifier using RBF model.

3.1. End-Point Detection and Normalization

Artificial neural network works well for the isolated words of speech signal in small vocabulary, but is week for the dynamic patterns. To recognize the speech signal, speech is transferred as static patterns by end-point detection and normalization in time and power. In this paper, end-point detection is materialized as Fig. 5. Word boundaries are detected

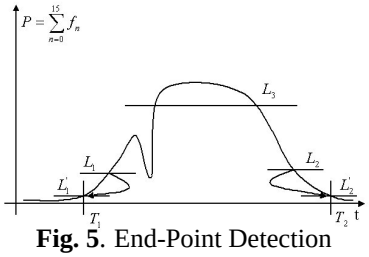


Fig. 5. End-Point Detection

by the intensity, which is the summation of ZCPA feature at each frequency bin. L_1 and L_2 are the rough ending point level over which the intensity is 8 times and L_1' and L_2' are the minimum levels of actual boundary. To avoid backward processing the system checks all levels($L1$ and L_1 or $L2$ and L_2), and finds the T_1 or T_2 first and determines whether the end point is true or not. Finally, the input signal is decided as a word if the power goes above L_3 at least 4 times. If the end-point detection is done, the feature vector is normalized by trace-segmentation algorithm in time domain, and normalized to a fixed intensity. The division operation required in this stage is implemented by

$$A/x = A \times M/2^8 \quad (4)$$

$$M = 2^8/x \quad (5)$$

where M is read from a look-up table. As an 8 bit multiplier is necessary in section 3.2, we use the order of 8. In this paper the input vector of classifier is normalized as a 16x64 static pattern.

3.2. RBF Networks

Fig. 6 shows the network architecture of RBF classifier. The

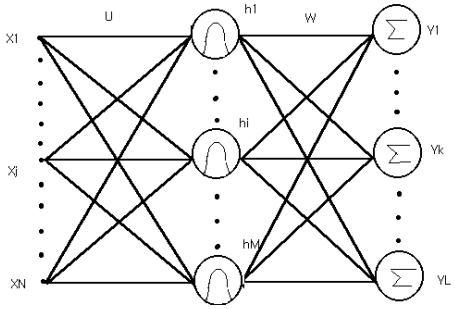


Fig. 6. Radial Basis Function Model

number of input speech features is set to $16 \times 64 = 1024$. The RBF model is operated by

$$h_{ij} = \exp\left[\sum_{j=0}^{1023} -D_{ij}^2/(2 \times 1024 \times \sigma_{ij}^2)\right] \quad (6)$$

$$D_{ij}^2 = (x_j - u_j)^2 \quad (7)$$

$$y_k = \sum_i h_i w_{ki} \quad (8)$$

To calculate these values with limited dynamic ranges of integer computation, we use

$$h_i = \exp[t_i/64] \quad (9)$$

$$t_i = \sum_{i=0}^{1023} D_{ij}^2/\left(\sigma_{ij}^2\right) \quad (10)$$

$$D_{ij}^2 = (x_j - u_j)^2/32 \quad (11)$$

$$y_k = \sum_i h_i w_{ki} \quad (12)$$

Here, x_j and u_j denote the jth component of the input feature and mean vector of the ith hidden neuron. Each hidden neuron has its unique variance value σ_{ij}^2 for each input component. Equation 9 is implemented as a look-up table. Based on these formula, we design the classifier which shows 95.2 % correct recognition rate in 8 bit depth and 95.6 % in 16 bit depth. In out chip, the neural network classifier is constructed as 8 bit multiplier which has signed and unsigned operations. The normalization stage in section 3.1 needs unsigned operation for maximum resolution of ZCPA feature vector, that has 8 bit depth. In our chip design the multiplier for signed and unsigned operation is implemented by the Pezaris multiplier[4].

3.3. Learning and Adaptation

In the phase of learning, the floating point computation is needed to express the difference between target and weight. The learning is executed at workstation but the speaker adaptation is accomplished in the chip. The training parameters of RBF classifier are means and variances of hidden neurons. The means are adapted by just add and shift as $M_{new} = M_{old} + (F_{input} - M_{old}) >> 4 = 0.94 \times M_{old} + 0.06 F_{input}$ where F_{input} is the feature vector with the highest firing rate of output neuron, and variances are adapted from means by a look-up table. The performance is improved from 95.2 % to 98.0 % via adaptation.

3.4. Rejection

For real-world applications it is also important to reject OOV words. To improve rejection performance without reducing recognition rates is very difficult. Although RBF networks demonstrated good rejection performance, we provided an additional constraint. With the help of simple finite-state logic, we let the chip actually execute the command only if a specific pre-command word proceeds the actual command. The pre-command word may be entered at the boot sequence. As a result, the classifier for pre-command is the speaker dependent, while actual commands are independent.

4. SYSTEM CONFIGURATION AND RESULTS

Fig. 7 shows the designed board. The system is composed of an amplifier, Sample and Hold, 12 bits Analog to Digital Converter, 128 KBytes SRAM and ROM, and main speech recognition processor. The main processor, ZCPA feature extractor and RBF classifier, is embedded in Altera Flex10K and uses about 160 thousand gates. By 2.1.1 the system needs 12 MHz oscillator and the input signal is sampled as 11.025 KHz with 12 bit quantization level. It takes 15 ms to recognize the command after end-point detection. The recognition rate is improved to 98 % with speaker adaptation and pre-command.

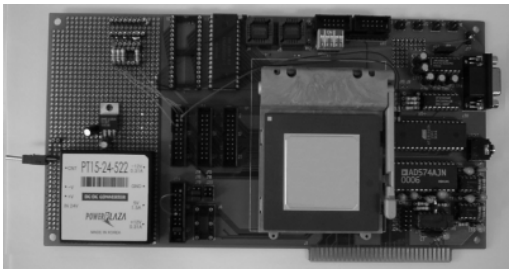


Fig. 7. System Photograph

5. CONCLUSIONS AND FURTHER WORKS

The isolated word recognition system for real world application is proposed and implemented by FPGA. For the robustness to background noise, we use the ZCPA model and propose the modified algorithm for digital chips. For both the rejection to OOV inputs and speaker adaptation, we used RBF model for classifier. The developed chip shows 98 % recognition rate for 50 isolated Korean words with speaker adaptation and pre-command. The chip may operate speaker independent, speaker dependent and/or speaker speaker adaptation mode. The system is currently under development as a single ASIC.

6. REFERENCES

[1] D.S. Kim, S.Y. Lee, and R.M. Kil, "Auditory Processing of Speech Signals for Robust Speech Recognition in Real-World Noisy Environments., *IEEE Trans, Speech and Audio Processing*, vol. 7, p55-69. Jan. 1999.

[2] O. Ghitza, "Auditory models and human performances in tasks related to speech coding and speech recognition", *IEEE Trans. Speech Audio Signal Processing*, vol. 2, pt. II, p. 115-132, 1994.

[3] G. von. Bekesy, *Experiments in Hearing*, McGraw-Hill, 1960.

[4] Stylianos D. Pezaris, "A 40-ns 17-Bit by 17-Bit Array Muliplier", *IEEE Trans. Computers*, vol. 2, no. 3, p. 421-435, 1971.

[5] Y. C. Lim and S. R. Parker, "FIR Filter Design Over a Discrete Powers-Of-Two Coefficients Space.", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31 p 583-591, June 1983.