(IJIEST) 2020, Vol. No. 6, Jan-Dec **e-ISSN: 2454-9584; p-ISSN: 2454-8111**

VLSI Design for SVM-Based Speaker Verification System

Sreemathy T G
Lecturer in Electronics,
Govt Polytechnic College, Kunnamkulam

ABSTRACT

This brief presents the chip implementation of a support vector machine (SVM)-based speaker verification system. The proposed chip comprises a speaker feature extraction (SFE) module, an SVM module, and a decision module. The SFE module performs autocorrelation analysis, linear predictive coefficient (LPC) extraction, and LPC-to-cepstrum conversion. The SVM module includes a Gaussian kernel unit and a scaling unit. The purpose of the Gaussian kernel unit is first to evaluate the kernel value of a test vector and a support vector. Four Gaussian kernel processing elements (GK-PEs) are designed to process four support vectors simultaneously. Each GK-PE is designed in the pipeline fashion and is capable of performing 2-norm and exponential operations. An enhanced CORDIC architecture is proposed to calculate the exponential value. As well as the Gaussian kernel unit, a scaling unit is also developed for use in the SVM module. The scaling unit is used to perform scaling multiplications and the remaining operations of SVM decision value evaluation. Finally, the decision module accumulates the frame scores that are generated by all of the test frames, and then compare it with a threshold to see if the test utterance is spoken by the claimed speaker. This designed chip is characterized by its high speed and its ability to handle a large number of support vectors in the SVM. The prototype chip is a semicustom chip that is fabricated using Taiwan Semiconductor Manufacturing Company 0.90-nm CMOS technology on a die with a size of - 7.9 mm X 7.9 mm.

Keywords:- *VLSI Design; SVM; CORDIC; Gaussian kernel; CMOS*

INTRODUCTION

A biometric system makes a pattern recognition decision in accordance with the biometric features extracted from a human being. In recent years, various human characteristics such as the face, speech, fingerprint, and iris have been considered as discriminative features for automatic biometric recognition. In this brief, it is addressed on hardware design of a speech based biometric system, i.e., speaker recognition system. Basically, speaker recognition systems are divided into two main categories: speaker identification and speaker verification. In a speaker identification system, an unknown speaker is identified as one of the speakers in the database. In a speaker verification system, a person's identity is validated based on his/her speech feature.

Speaker recognition has been extensively studied for the last decades. Feature extraction and classifier design are the two essential issues in a speaker verification system. For the feature extraction, the most frequently adopted speaker features are cepstral coefficients. The cepstral coefficients can be extracted by two dominant approaches. One is the parametric approach, which is developed to match closely the resonant structure of the human vocal tract that produces the corresponding speech sound. This approach is mainly based on linear predictive analysis. The obtained linear predictive coefficients (LPCs) can be converted to LPC cepstral coefficients (LPCCs). The other one is the nonparametric method modeling the human auditory perception system. Mel frequency cepstral coefficients (MFCCs) are used for this purpose. Recently, a feature called the supervector, which is derived by concatenating the mean vectors of the components of a Gaussian mixture model (GMM), has received considerable interest. The dimensionality of a supervector typically exceeds 10000. Besides, the supervector requires higher computational load and larger buffer size than cepstral coefficients. Therefore, acoustical LPCC features are used herein instead of supervectors. In the classifier design, modern speaker recognition systems apply GMMs. The widespread use of GMMs in modeling speakers is based on the efficient parameter estimation procedures that involve maximizing the

likelihood of the model data. However, as a maximum likelihood- derived decision surface is not optimal, discriminative approaches are essential for creating robust and more accurate models. The support vector machine (SVM), a discriminative approach, has recently attracted much attention because it discriminates between the classes and can be used to train nonlinear decision boundaries efficiently.

Various hardware designs for speech or speaker recognition systems have been presented. However, in most of the relevant works, conventional recognition algorithms such as neural net- work, hidden Markov model, and dynamic time warping have been implemented. Hardware implementations for modern SVM-based speech or speaker algorithms are few. Manikandan et al. developed a speech recognition system using TMS320C6713 floating point digital signal processor. This system adopted SVM as the recognition engine and was capable of recognizing isolated digits. However, this system used a general purpose DSP chip to program the speech recognition algorithms. A text-independent SVM-based speaker identification system was developed. In this investigation, VLSI architecture of a sequential minimal optimization (SMO) algorithm for SVM learning was focused. The proposed VLSI architecture consisted of three modules and was tested using a Cyclone II 2C70 field programmable gate array (FPGA). The disadvantage of this paper is that the feature extraction and speaker identification processes were both performed in PC. Another work further extended the VLSI SMO and presented a hardware and software co-design solution for a fast-trainable speaker identification system. The proposed system consisted of a training phase and a multiclass identification phase. The SMO training algorithm was realized as a dedicated VLSI module, i.e., the hardware component. The feature extraction and SVM voting strategy were implemented by software. This system was implemented on a Socle CDK platform with an AMBA-based Xilinx FPGA and an ARM926EJ processor. However, this paper used linear kernel in SVM. Although the heavy computational load in evaluating a decision function can be avoided using a compaction technique introduced by the use of the linear kernel, the recognition performance of the linear kernel is much worse than that using the Gaussian kernel. Furthermore, the aforementioned previous works either used general-purpose DSP or used FPGA plus processor to implement SVM-based speech or speaker recognition algorithms. The cost of the resulting product would be high and power dissipation would be a problem. Particularly, low power dissipation is a key requirement for portable applications.

In this brief, an application-specific integrated circuit (ASIC) chip is presented for SVM- based speaker verification. The reason that an ASIC solution is selected is ASIC chips have the following advantages over FPGAs, DSPs, and microprocessors. First, the power dissipation of the ASIC is lower. Second, the cost of the resulting product is lower. Third, the speaker feature extraction (SFE) module and an SVM module were implemented as intellectual property (IP) cores, which can be easily reused and incorporated into designs of other systems or chips. The proposed chip is used in the speaker verification phase, after the training phase has already been completed using a PC. To ensure satisfactory verification performance, the Gaussian kernel is used in the SVM engine. The proposed chip design is expected to be used in the applications that require both the outputs of speech and speaker recognition, such as conversational spoken dialog systems and personal command systems. In addition, to improve speech recognition performance, speaker recognition can be used as a front-end process to select the most appropriate speech recognizer. In such systems, speech content and speaker's identity are recognized simultaneously.

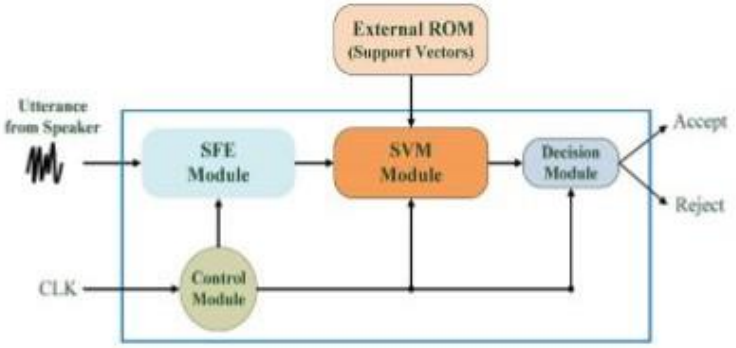

Figure 1.Block diagram of proposed speaker verification chip.

SYSTEM OVERVIEW

The block diagram of the proposed SVM-based speaker verification chip is shown in Fig.1. The SFE procedure is the same for both the training and the verification phases. First, speech data is preemphasized through a low order digital system. The preemphasized speech data are then blocked into overlapping frames so that smooth LPC spectral estimates can be obtained. After applying a window function to the speech frame, autocorrelation and linear predictive analyses are performed to obtain the LPC coefficients. Then, the cepstral coefficients are obtained by converting the LPC coefficients. The obtained cepstral coefficients are called LPCC or LPC cepstrum and are used as the speaker features in this brief.

The SVM is used to achieve speaker verification. In the enrollment phase, the hyperplane of a 2-class SVM is learned with the target side against the background side. Each enrolled speaker corresponds to a 2-class SVM, and constructs target side by collecting his own training feature vectors. The background side, which is the same for all of the enrolled target speakers, is constructed by collecting training feature vectors from a large number of speakers. In the verification phase, an utterance from an unknown speaker is first transformed into a sequence of test feature vectors. Each test vector is fed to the SVM to generate the frame score. The frame scores for all of the test vectors are summed to yield an overall score to determine whether the test utterance was spoken by the claimed target speaker or not. The proposed speaker verification chip consists mainly of a SFE module, an SVM module, a decision module, and a control module. The advantage of the core-based design for SFE and SVM is its flexibility that allows the designer to develop a new system in short time using suitable IP cores. For the SFE module, since the adjacent frames are overlapping, an intelligent architecture is used to perform the autocorrelation analysis without the use of a huge buffer. This architecture allows the input buffer for the SFE module to contain only two registers in- stead of a huge buffer. The SVM module includes a Gaussian kernel unit and a scaling unit. For each test frame, the Gaussian kernel unit computes the kernel values of a test vector and all of the support vectors using four processing elements. Based on these obtained kernel values, the scaling unit completes the remaining operations of decision function evaluation to output the frame score. This procedure is repeated for all of the frames. The decision module can, thereby compute the overall score by summing all of the frame scores. This obtained overall score is compared with a threshold to determine whether the test utterance is spoken by the person who claimed to have spoken it.

ARCHITECTURE OF SVM MODULE

A. SVM Theory

The SVM theory is a statistical technique and has drawn much attention on this topic in recent years. An SVM is a binary classifier that makes its decisions by constructing an optimal hyperplane that separates the two classes with the largest margin. It is based on the idea of structural risk minimization induction principle that aims at minimizing a bound on the generalization error, rather than minimizing the mean square error. For the optimal hyperplane $w \cdot x + b = 0$, $w \in R^N$ and $b \in R$, the SVM decision function for classifying an unknown point x is defined as

$$f(\mathbf{x}) = \mathbf{w}\mathbf{x} + b = \sum_{i=1}^{N_S} \alpha_i t_i \mathbf{y}_i \cdot \mathbf{x} + b$$

where N_S is the support vector number; y_i is the ith support vector; αi is the corresponding Lagrange multiplier; and $t_i \in \{-1, +1\}$describes, which class y_i belongs to. In most of the cases, searching suitable hyperplane in input space is too restrictive to be of practical use. The solution to this situation is mapping the input space into a higher dimension feature space and searching the optimal hyperplane in this feature space. Let $z = \varphi(x)$ denote the corresponding feature space vector with a mapping φ from R^N to a feature space Z. It is not necessary to know about φ. We just provide a function k (. , .) called kernel, which uses the points in input space to compute the dot product in feature space Z, that is

$$\varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{y}_j) = k(\mathbf{x}_i, \mathbf{y}_j).$$

Finally, the SVM decision function becomes

$$f(\mathbf{x}) = \sum_{i=1}^{N_S} \alpha_i t_i k(\mathbf{y}_i, \mathbf{x}) + b.$$

Functions that satisfy Mercer's theorem can be used as kernels. Typical kernel functions include linear kernel, polynomial kernel, and Gaussian kernel.

B. Proposed SVM Module

The purpose of the SVM module is to evaluate the SVM decision function for a test point, which is a 10-D LPCC vector. According to the equation just above, kernel evaluations must be carried out at an unknown test point x with all of the support vectors that were obtained by SVM hyperplane training. To ensure satisfactory classification performance, a large number of support vectors must generally be used. Completing all of the kernel evaluations is thus a computationally intensive task. Therefore, an efficient SVM computing architecture is essential so that both real-time speaker verification and low power dissipation can be accomplished.

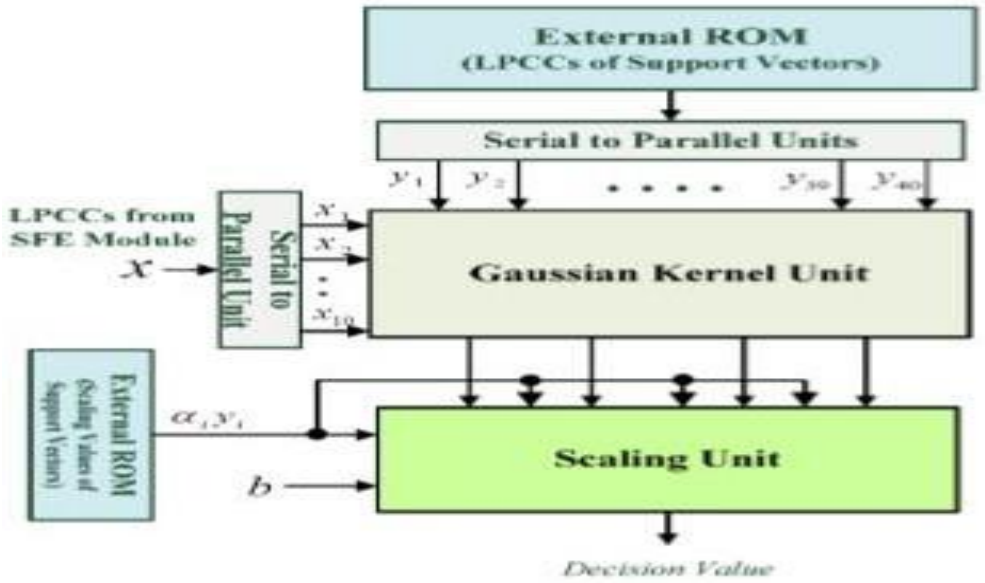

Figure 2: Block diagram of SVM module

The proposed system adopts the Gaussian kernel, which performs excellently in the simulation. The Gaussian kernel is defined by

$$k(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}}$$

where x and y are the two vectors whose kernel value is evaluated, and σ is the standard deviation.

In this SVM-based speaker verification, each frame forms a test vector and each test vector performs kernel evaluations with all of the support vectors. As stated previously, the last two equations impose a heavy computational load so two specific architectures are designed to compute them herein. Fig.2 shows the block diagram of the proposed SVM module, which mainly comprises of two computational units, a Gaussian kernel unit and a scaling unit, which calculate the equations.

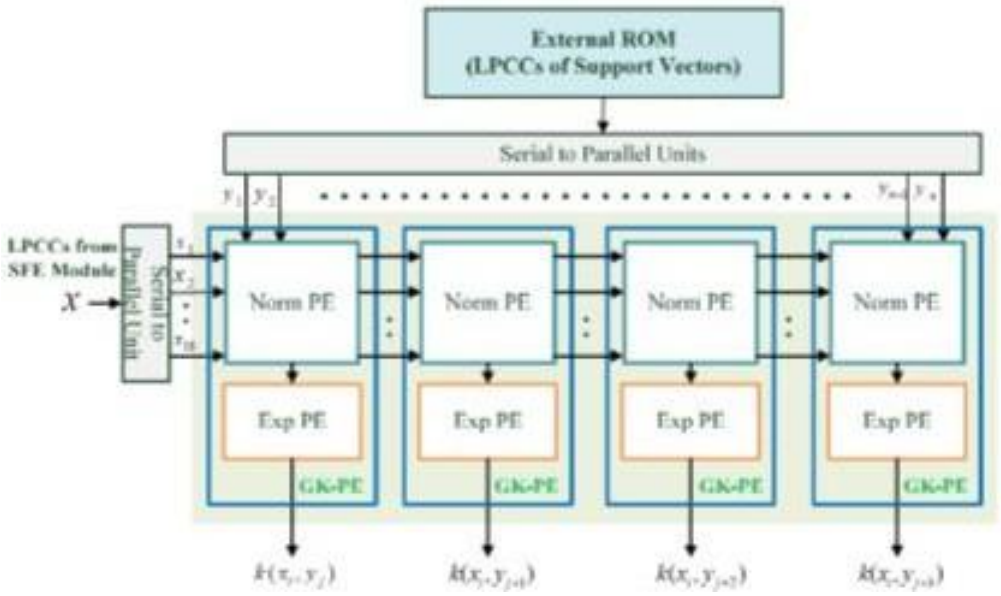

Figure 3: Architecture of Gaussian kernel unit

C. Gaussian Kernel Unit

The Gaussian kernel unit is designed to perform the Gaussian kernel evaluations of a test vector and all support vectors. Fig. 3.2 shows the proposed architecture of the Gaussian kernel unit, which consists of four Gaussian kernel processing elements (GK-PEs) and ïň Ąve serial- to-parallel units (SPUs). For a test vector, each GK-PE performs its Gaussian kernel evaluation with one of the support vectors. Four GK-PEs are adopted in the Gaussian kernel unit so four support vectors can be processed simultaneously.

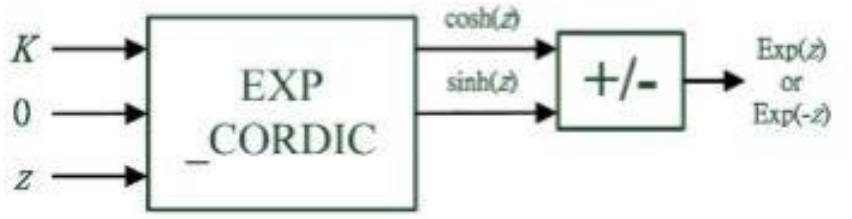

Figure 4: Block diagram of the Exp PE

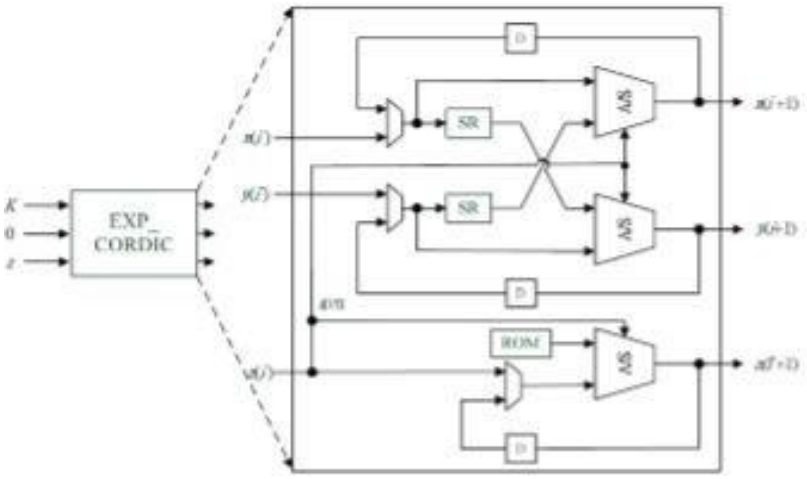

Figure 5: CORDIC circuit

(IJIEST) 2020, Vol. No. 6, Jan-Dec **e-ISSN: 2454-9584; p-ISSN: 2454-8111**

The above-mentioned SFE module generates each dimension of an LPCC vector sequentially. An SPU is used to transform the serially input data to data that are input in parallel for GK-PE. One SPU receives the test vector while each of the other four SPUs takes its corresponding support vector. Each GK-PE incorporates a norm PE (Norm PE) and an exponential PE (Exp PE). The architectures of the two PEs are described in detail as follows.

Norm PE: The Norm PE is responsible for calculating $-||(x-y)||^2/2\sigma^2$, where x = (x1, x2, ... , x10) denotes an LPCC test vector, and y = (y1, y2, ... , y10) represents an LPCC support vector. The Norm PE ïň Ąrst computes the norm square, $||x-y||^2 = (x1-y1)^2 + (x2-y2)^2 + ... + (x10-y10)^2$, with an adder tree to sum the square values of difference in each dimension. The standard deviation is taken as one. Hence, only the compliment of a 2 and a right-shift operation are required to obtain $-(x-y)^2/2\sigma^2 \; from \; ||x-y||^2$.

(1) Exp PE: Among the several hardware implementation options in exponential operation, the CORDIC method is used. The CORDIC usually occupies a small area of hardware because it merely uses several adders and shifters. With the unfolding technique, CORDIC can achieve high speed.

The Exp PE in Fig. 4 has a CORDIC circuit and an adder/subtractor. The constant K is precalculated and stored in memory. Fig 5 shows the CORDIC circuit, which is capable of performing the angle updating operations. One ROM is used in the CORDIC circuit to store the precalculated $tanh^{-1}(2^{-i})$. In the CORDIC circuit, more iterations correspond to higher numerical precession in cos h(z) and sin h(z). Simulation results reveal that 13 iterations yield a satisfactory numerical precession.

However, the above circuit can only handle the input z with a value of between -1 and 1. With a z above this range, a large error arises in the resulting exponential value. To solve this problem, a numerical transformation is provided. Any number $z \in R^{+}$

$$z = z_1 + p \ln 2$$

can be expressed as where $p \in Z^{+} and z1 \in [-1,1]$.

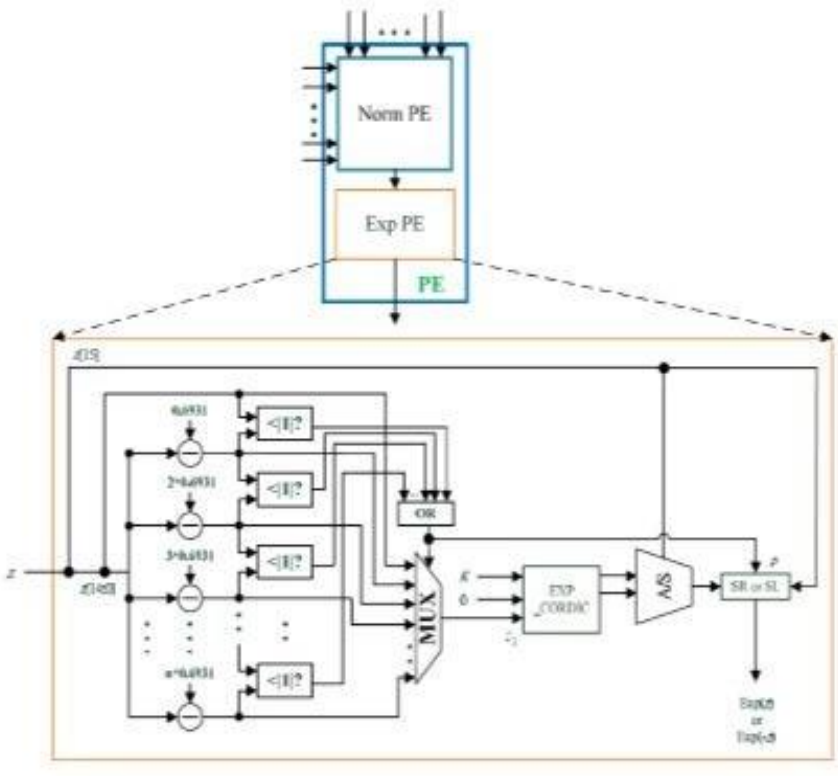

Figure 6: Architecture of Exp PE

$$e^{z} = e^{z1+p\ln 2} = e^{z1} \cdot e^{p\ln 2} = 2^{p} \cdot e^{z1}.$$

Accordingly, performing the exponential operation on Z yields
Based on the above equation, e^{z} is replaced by e^{z1} with $z1 \in [-1, 1]$, and then a right shifting by P bits is conducted. Accordingly, the architecture of Exp PE is developed as shown in Fig. 3.5.

D. Scaling Unit

The first task in the scaling unit is to multiply each k(xi, x) that is generated from the Gaussian kernel unit by the corresponding scaling coefficient, αiyi, which is obtained in the training phase and stored in ROM. In line with the Gaussian kernel unit, four multipliers perform the scaling multiplications on the four kernel values that are received from the Gaussian kernel unit in parallel. A two-stage adder tree is used to sum these four multiplications. An accumulator is adopted to accumulate each of the four scaling multiplication results until the kernel values of all of the support vectors have been scaled. Finally, the stored bias constant b is added to the sum of the results of the scaling multiplication, $\Sigma^{n}_{i=1} \alpha itik(x, yi)$, to generate the SVM decision value. The decision value of each LPCC test vector is sent to the decision module to make an overall decision.

Implementation Results

The speaker verification system using MATLAB software is built. The sampling rate of the input speech is 16 KHz. The frame size is 256 samples with 128 sample frame overlapping.

The required finite word length accuracy is analyzed first by software simulation, allowing for implementation of the floating-point program by a fixed-point structure. The design uses a 16-bit fixed-point format, which comprises four bits for the integer part and 12 bits for the fractional part. To evaluate the performance of the speaker verification system, the experiment is conducted on spoken data of 200 speakers were taken from the NIST SRE database. In this experiment, three numbers of support vectors - 3000, 6100, and 12500 were used to train the speaker model. The equal error rates (EERs) that were generated using 3000 and 6100 support vectors were approximately 8.27% and 7.92%, respectively. A total of 12500 support vectors yielded an EER of around 7.51%. The experimental results reveal that using more support vectors may slightly improve the EER. The speaker verification performance of the presented VLSI design is satisfactory and close to that of other SVM-based speaker verification systems that also use acoustical speech features such as MFCC.

The chip design was implemented using the Taiwan Semiconductor Manufacturing Company (TSMC) 0.90 nm CMOS technology and the cell-based design ï ň Ć ow. Hardware simulation of the proposed chip architecture was conducted by Verilog HDL where a prototype chip was designed using Cadence's front- and back-end tools. Fig. 4.2 shows the layout view of the chip. The total gate count is around 1731 K with a die size of roughly 7.9mm X 7.9 mm. With a power supply of 0.9 V, the design can achieve 100 MHz in the worst case; in addition, the power dissipation is roughly 8.12 mW at this speed. Low power dissipation makes this chip appropriate for portable applications. Table I summarizes the chip specifications. The execution time of our chip depends on the support vector number. For a speaker model with 12500 support vectors, the required clock number to process a speech frame (feature extraction + classification) is about 234830 clock cycles. At a clock rate of 100 MHz, the proposed chip takes 2.35 ms to process a speech frame. The chip is capable of performing real-time verification as a new speech frame is captured each 8 ms.

TABLE I
SPECIFICATION OF PROPOSED CHIP

CMOS Technology	TSMC 0.90nm Standard Cell Library
Gate Count Number	1731K
Max. Clock Rate	100 MHz
Power Consumption	8.12mW@1V, 100MHz
Core Size	4.42x4.42 mm^2
Die Size	7.9x7.9 mm^2

Figure 7: Chip Specification

CONCLUSION

In this brief, a pure AISC solution to the SVM-based speaker verification is presented. This ASIC chip is characterized by its modular design, high speed, and low power. The architecture consists of a SFE module, an SVM module, and a decision module. The SFE module yields the LPC cepstrum vector. The SVM module evaluates all of the required kernel values, performs scaling multiplications, and completes the remaining operations of decision value evaluation. The decision module computes the overall score of a test utterance to make an accepting or rejection decision. With TSMC 0.90 nm CMOS technology, the die size of the proposed ASIC chip is roughly $7.9X7.9mm^2$. For a 0.9 V power supply, the maximum clock rate is 100 MHz and the power dissipation is $\sim 8.12mW$. The proposed speaker verification ASIC can be used alone or integrated with other biometric chips and peripheral components to form a multimodal biometric system on a chip.

REFERENCES

1. **B. H. Juang and T. H. Chen** The past, present, and future of speech processing. In *IEEE Signal Process. Mag., vol. 15, no. 3, pp. 24-48 (May 1998)*

2. **J. C. Wang, C. H. Yang, J. F. Wang, and H. P. Lee** Robust speaker identification and verification In *IEEE Comput. Intell. Mag., vol. 2, no. 2, pp. 52-59, (May 2007)*.

3. **W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, and P. A. Torres- Carrasquillo** Support vector machines for speaker and language recognition In *Comput. Speech Lang., vol. 20, nos. 2-3, pp. 210-229, (2006).*

4. **C. H. Yang, J. C. Wang, J. F. Wang, C. H. Wu, and F. M. Li** VLSI architecture and implementation for speech recognizer based on discriminative Bayesian neural network In *IEICE Trans. Fundam. Electron., Commun. Comput. Sci., vol. E85-A, no. 8, pp. 1861-1869, (Aug. 2002.)*.

5. **J. F. Wang, J. C. Wang, H. C. Chen, T. L. Chen, C. C. Chang, and M. C. Shih** Chip design of portable speech memopad suitable for persons with visual disabilities In *IEEE Trans. Speech Audio Process., vol. 10, no. 8, pp. 644-658, (Nov. 2002)*.

6. **V. Vapnik** Statistical Learning Theory In *New York, NY, USA: Wiley, (1998).*