Sub-Microwatt Analog VLSI Trainable Pattern Classifier

Shantanu Chakrabartty, *Member, IEEE*, and Gert Cauwenberghs, *Senior Member, IEEE*

***Abstract*—The design and implementation of an analog system-on-chip template-based pattern classifier for biometric signature verification at sub-microwatt power is presented. A programmable array of floating-gate subthreshold MOS translinear circuits matches input features with stored templates and combines the scores into category outputs. Subtractive normalization of the outputs by current-mode feedback produces confidence scores which are integrated for category selection. The classifier implements a support vector machine to select programming values from training samples. A two-step calibration procedure during programming alleviates offset and gain errors in the analog array. A 24-class, 14-input, 720-template classifier trained for speaker identification and fabricated on a 3 mm × 3 mm chip in 0.5 μm CMOS delivers real-time recognition accuracy on par with floating-point emulation in software. At 40 classifications per second and 840 nW power, the processor attains a computational efficiency of 1.3×10^{12} multiply-accumulates per second per Watt of power.**

***Index Terms*—Micropower techniques, machine learning, biometrics, MOS translinear principle, flash analog memory, smart sensors, vector ADC.**

I. Introduction

ENERGY efficiency of information processing is a key design criterion in the development of ultra-low-power autonomous sensors. By embedding information extraction capability directly at the sensor interface, the communication bandwidth requirement at the sensor can be relaxed, leading to significant savings in power. Fully autonomous sensors capable of scavenging energy from the environment to perform information processing and communication are being pursued by several groups [1]–[3]. Current energy harvesting techniques (other than solar energy) are limited to less than 10 μW of continuous power [4], motivating the design of systems operating within sub-microwatt power budgets. Amirthrajah *et al.* [5] have recently shown the feasibility of a sub-microwatt digital signal processor using "approximate processing" techniques where precision in computation is traded off with power consumption. An attractive alternative to digital signal processing (DSP) is analog signal processing (ASP), which utilizes computational primitives inherent in device physics to achieve high energy and integration efficiency [6], [7]. For instance, ASP techniques have been used to implement recognition systems for biomedical sensors [8], [9] and for sequence identification in communications [10], [11]. The use of ASP relaxes the precision requirement on analog-to-digital conversion (ADC) which typically dominates the power dissipation of a DSP-based sensor. However, imperfections in analog VLSI implementation due to noise, mismatch, offset and distortion [12] limit the precision that can be attained by ASP. Any ASP-based technique therefore has to provide a principled approach to compensate for such imperfections.

This paper describes an implementation of a sub-microwatt analog pattern recognition system-on-chip for biometric signature verification. While the pattern classifier applies generally to other types of signals, we chose a speaker verification task as a proof of principle demonstration. Fig. 1 shows the system architecture with acoustic features supplied to the pattern classifier by an acoustic front-end. The confidence scores generated by the classifier are integrated over the duration of the acoustic event to verify presence of a pattern of interest. The classifier is trained as a support vector machine (SVM). SVMs have been applied successfully to several demonstrating excellent generalization ability even with sparse training data [13], [14]. SVM classification is attractive for analog VLSI implementation because it lends itself to an array-based solution with a high degree of regularity in computation [15]. The SVM classifier here has been implemented as a current-mode array of floating gate MOS translinear circuits with embedded analog storage. High energy efficiency is achieved by biasing MOS transistors in the subthreshold region, where power–delay products are minimized and are constant over several decades of operating range [6], [7]. The chip is fully programmable, with parameter values from SVM training downloadable onto the floating gate array. Using calibration and chip-in-loop training, imperfections due to mismatch and nonlinearity in analog implementation are alleviated.

Although the chip is designed to implement adaptive dynamic sequence identification [16] according to a kernel-based forward decoding algorithm [17], the architecture applies generally to template-based pattern recognition, and we present the micropower operation of the chip with its application to biometric verification.

This paper is organized as follows. Section II presents the classification architecture in the context of the speaker verification system. Section III describes the circuit implementation of the classifier, and Section IV expresses fundamental limits of the basic building blocks. Section V presents the fabricated prototype and calibration procedures to compensate for analog imperfections. Experimental results with the system-on-chip con-

Manuscript received April 4, 2006; revised December 5, 2006. This work was supported by a grant from The Catalyst Foundation (http://www.catalyst-foundation.org), NSF IIS-0209289, NSF IIS-0434161, and ONR/DARPA N00014-00-C-0315. The chip was fabricated through MOSIS.

S. Chakrabartty is with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: shantanu@msu.edu).

G. Cauwenberghs is with Section of Neurobiology, Division of Biological Sciences, University of California at San Diego, La Jolla, CA 92093 USA.

Digital Object Identifier 10.1109/JSSC.2007.894803

0018-9200/$25.00 © 2007 IEEE

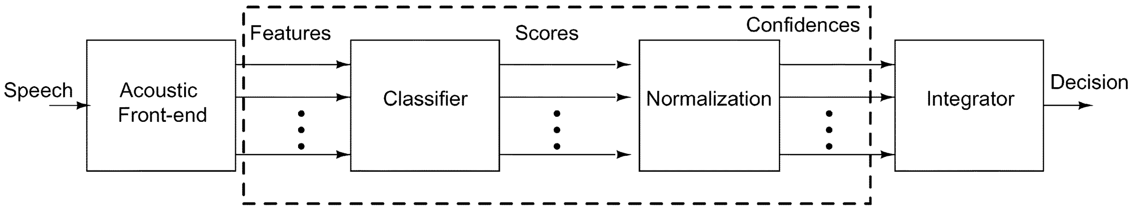


Fig. 1. System diagram of acoustic classifier for speaker verification.

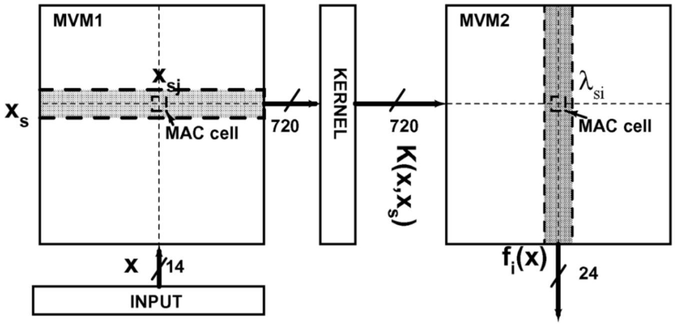


Fig. 2. Parallel architecture implementing multi-class SVM classification.

figured for speaker verification task are given in Section VI. Section VII concludes with final observations.

II. Classification Architecture

The acoustic front-end in Fig. 1 generates speech feature vectors $\mathbf{x}[n] \in \mathcal{R}^d$ at discrete time instances $n = 1, \ldots, N$. Each sample $\mathbf{x}[n]$ in the sequence is presented to the classifier, which implements a support vector machine (SVM)-based classification algorithm. SVMs were originally formulated for binary classification [13] but extend to multi-class or multi-category classification [14]. The SVM classifier computes matching scores ("kernels") between the input vector $\mathbf{x}[n]$ and a set of S template vectors ("support vectors") $\mathbf{x}_s \in \mathcal{R}^d (s = 1, \ldots, S)$. It then linearly combines these scores to produce outputs f_i for each category ("class") $i = 1, \ldots, M$ according to

$$f_i(\mathbf{x}[n]) = \sum_{s=1}^{S} \lambda_{si} \mathbf{K}(\mathbf{x}_s, \mathbf{x}[n]) + b_i. \tag{1}$$

A quadratic kernel $K(\mathbf{x}_s, \mathbf{x}) = (\mathbf{x}_s \cdot \mathbf{x})^2$ is implemented, satisfying the Mercer condition for convergence of SVM training [14]. The SVM training procedure automatically selects support vector templates $\mathbf{x}_s$ from the training examples, and derives values for the coefficients λ_{si} and offsets b_i accordingly.

A parallel architecture implementing the multi-class SVM (1) is shown in Fig. 2, containing two matrix vector multipliers MVM1 and MVM2. The input vector $\mathbf{x}[n]$ is presented column-parallel to MVM1, which computes inner-products between the template vectors $\mathbf{x}_s$ and the input vector $\mathbf{x}[n]$. With unsigned (non-negative) acoustic features produced by the front-end, the input vectors $\mathbf{x}[n]$ as well as the support vectors $\mathbf{x}_s$ selected from its training samples have all unsigned components, so that all MAC operations in MVM1 reduce to single-quadrant, conveniently implemented with translinear current-mode circuits (Section III). The inner-products returned by MVM1 are squared to produce the kernel values $K(\mathbf{x}_s, \mathbf{x}) = (\mathbf{x}_s \cdot \mathbf{x})^2$. The kernels are presented row-parallel to $MVM2$ which computes the SVM category outputs f_i according to (1).

The scores f_i generated by the SVM are normalized to produce confidence scores, or measures of probability P_i for each category [17]. Normalized confidence scores allow integration over the duration of the speech sequence $n = 1, \ldots, N$ for reliable classification, and reduce false alarms due to outliers generated by the front-end.

A reverse water-filling algorithm, popular in rate distortion theory [21], implements a *subtractive* normalization procedure [18]. The normalized confidence values $P_i[n]$ are obtained from the classifier outputs $f_i (i = 1, \ldots, M)$ according to

$$P_i[n] = [f_i(\mathbf{x}[n]) - Z[n]]_+ / \eta \tag{2}$$

where $[x]_+ = \max(x, 0)$, and η is a constant normalization parameter. The threshold level $Z[n]$ is set by the reverse water-filling criterion

$$\sum_{i}^{M} [f_i(\mathbf{x}[n]) - Z[n]]_+ = \eta. \tag{3}$$

It readily follows that $P_i[n]$ represents a valid probability measure, $0 \leq P_i[n] \leq 1$ and $\sum_i P_i[n] = 1$. Compared to a more conventional method of *divisive* normalization $P_i[n] = f_i(\mathbf{x}[n]) / \sum_i^M f_i(\mathbf{x}[n])$ [19], [20], the following properties of the subtractive normalization are noteworthy.

- Subtractive normalization applies directly to multi-class SVM classification outputs f_i [18] whereas divisive normalization requires nonlinear transformation such as exponentiation to ensure positive arguments $f_i > 0$, more complex to implement accurately in analog circuits.
- The distribution P_i obtained by (2) is insensitive to uniform offset in f_i whereas the distribution P_i obtained by divisive normalization is insensitive to uniform scaling in f_i. Because $K(\mathbf{x}_s, \mathbf{x})$ is positive and identical for all classes i, the offset insensitivity in f_i allows to shift the parameters λ_{is} and b_i by any constants across i. The important implication is that these constants can be chosen sufficiently positive so that MVM2 in Fig. 2 implements (1) using just single-quadrant multipliers.
- For small values of the normalization parameter η the distribution $P_i[n]$ strongly favors the class with highest confidence $f_i(\mathbf{x}[n])$ and in the limit $(\eta \rightarrow 0)$ favors only

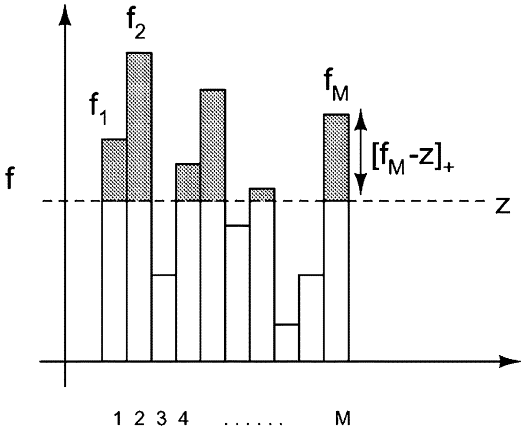


Fig. 3. Reverse water-filling procedure: the threshold Z is such that the net balance of scores $f_1, f_2, \ldots, f_M$ exceeding Z is fixed at η.

the class with highest confidence (winner-take-all) truncating the rest. In analog VLSI systems truncation is natural, and de-noising is achieved by adjusting the truncation threshold to the noise floor.

A pictorial description of the reverse water-filling procedure solving (3) is shown in Fig. 3 for scores corresponding to M classes $f_1, f_2, \ldots, f_M$. The threshold $Z[n]$ is obtained such that the combined score above the threshold (shown by the shaded area) equals η. Even though the reverse water-filling algorithm involves sorting and search techniques that would involve nested routines in digital implementation, in Section III we propose an analog VLSI network that directly solves for $Z[n]$, using (3) as its equilibrium criterion.

For each of the classes $i, n = 1, \ldots, N$ the corresponding normalized confidences $P_i[n]$ are integrated over the duration of the acoustic event. Event classification is performed by identifying the class $\hat{q}$ corresponding to maximum integrated confidence according to

$$\hat{q} = \arg\max_i \sum_n^N P_i[n]. \tag{4}$$

In a binary classification setting $i = 1, 2$ (e.g., speaker verification), an acceptance threshold θ is introduced in (4) to minimize the false acceptance rate. The acceptance of an acoustic event is determined by the sign of the binary decision function:

$$y = \text{sign}\left(\sum_n^N (P_1[n] - P_2[n]) - \theta\right). \tag{5}$$

III. Circuit Implementation

A. *Classifier Circuits*

The most computationally intensive operation of the SVM is matrix-vector multiplication (MVM) for which several implementations have been reported in literature [22], [15], [23]. In the present implementation each MVM is served by a floating gate MOS translinear array, each cell performing one multiply accumulate (MAC) operation.

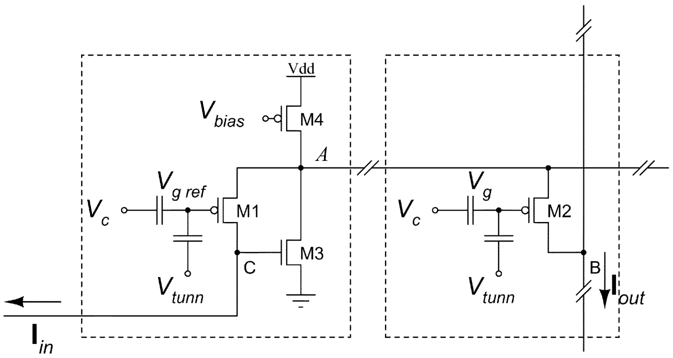


Fig. 4. Multiply accumulate cell (MAC, right) and reference cell (left) implementing MVM in Fig. 2.

Careful consideration at the architectural level, as presented in Section II, reduces every MAC operation in MVM1 and MVM2 to a single quadrant, with unsigned non-negative operands both for the input and the stored parameter in the multiplier. Unsigned multiplication and accumulation are conveniently implemented in the current domain by floating-gate MOS translinear circuits as follows. Applied to the MOS transistor operating in the subthreshold region [24], the translinear principle [25] makes use of the exponential relation between drain current I_{ds} and either gate voltage V_g or source voltage V_s (relative to bulk voltage) to implement products of currents as sums of voltages. In particular, for a pMOS transistor in subthreshold in saturation, the drain current

$$I_{ds} = I_o \frac{W}{L} e^{-\kappa V_g/U_T} e^{V_s/U_T} \tag{6}$$

decomposes in a product of two terms, the first exponential in V_g and the second exponential in V_s, on a voltage scale set by thermal voltage $U_T = kT/q \approx 26$ mV and gate coupling factor $\kappa \approx 0.7$ [7], [6], [29]. The circuit in Fig. 4 implements the first term by floating-gate storage, and the second term by pre-distortion of an input current I_{in} through current-mode feedback. High gain amplification by M3 and M4 sets the common-source voltage of M1 and M2 (node A in Fig. 4) such that M1 carries the input current I_{in}. The matched pair of transistors M1 and M2 then produces an output current

$$I_{\text{out}} = I_{\text{in}} e^{-\kappa(V_g - V_{\text{gref}})/U_T} \tag{7}$$
$$= I_{\text{in}} \beta \tag{8}$$

that is linear in the input current, scaled by floating-gate programmable weight $\beta = e^{-\kappa(V_g - V_{\text{gref}})/U_T}$, and independent of process parameters I_o and bulk voltage.

Two observations can be directly made regarding (8) and its circuit implementation in Fig. 4.

1) The input stage significantly reduces the effect of the bulk and any common-mode disturbance on the output current. This is illustrated in Fig. 5 where the common-mode disturbance is introduced by varying the control gate voltage V_c. The measured characteristic shows rejection of the common-mode variations by 20 dB in the output current.
2) The weight β is differential in the floating gate voltages $V_g - V_{\text{gref}}$, allowing to increase or decrease the weight by

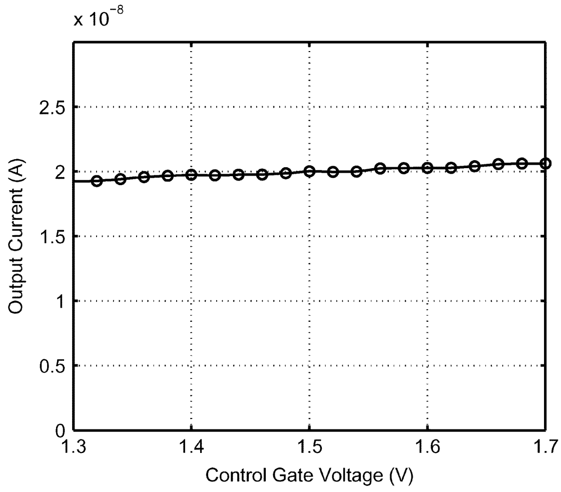


Fig. 5. Rejection of common-mode disturbance of the input stage.

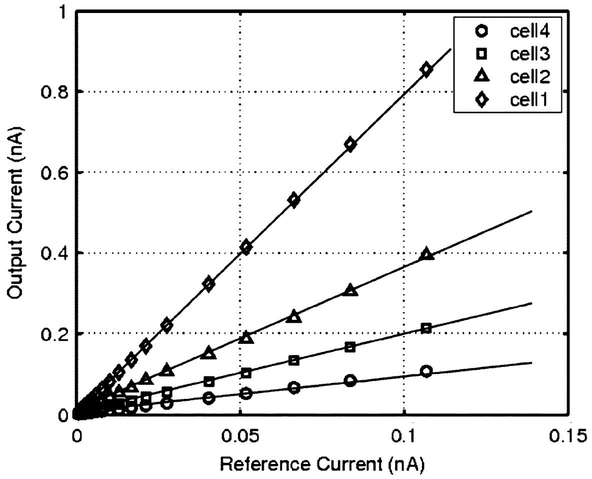


Fig. 6. Output current through four MAC cells programmed with currents that scale with ratios $1, 1/2, 1/4, 1/8$.

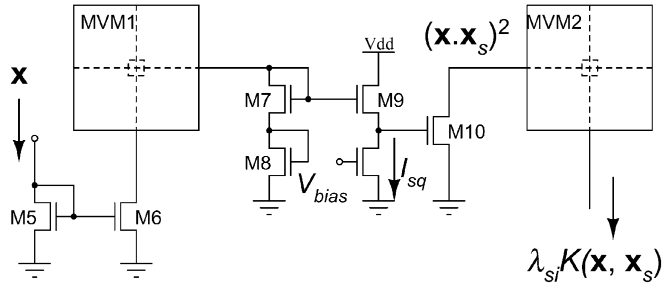

MVM1
x
M5
M6
M7
M8
V_{bias}
Vdd
M9
M10
I_{sq}
$(\mathbf{x}.\mathbf{x}_s)^2$
MVM2
$\lambda_{si}K(\mathbf{x}, \mathbf{x}_s)$

Fig. 7. Schematic of the SVM system with combined input, kernel and MVM modules.

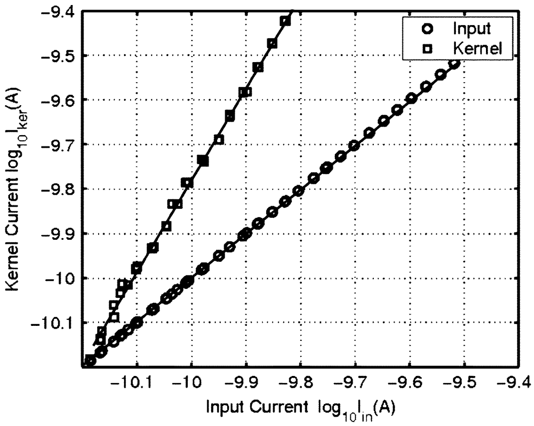


Fig. 8. Measured kernel characteristic showing approximate square dependence of output current on input current.

- hot electron injection only, without the need for repeated high-voltage tunneling. For instance, the leakage current in unused rows can be reduced significantly by programming the reference gate voltage to a high value, leading to power savings. The reference voltage V_{gref} is also used for compensating for the effects of mismatch between different rows of floating gate cells.

The feedback transistor in the input stage M3 reduces the output impedance of node A and is approximately given by $r_o \approx g_{d1}/g_{m1}g_{m2}$ where g_{m1}, g_{m2}, and g_{d1} are source and drain referred transconductances of $M2$ and $M1$. The low impedance at node A makes the array scalable, as additional memory elements can be added to the node without significantly loading the node. An added benefit of keeping the voltage at node A fixed is reduced variation in back gate parameter κ in the floating gate elements [27], [28]. The current from each memory element is summed on a low impedance node established by two diode connected transistors M7–M10 as shown in Fig. 7. This partially alleviates large drain conductance due to capacitive gate-drain coupling implicit in floating gate transistors [27]. Fig. 6 shows the output current through MAC cells whose floating gate cells have been programmed in a geometric fashion, demonstrating multiplication operation.

The subthreshold characteristics of electrode-coupled floating-gate MOS transistors supports the implementation of a linear matrix-vector multiplier [23]. Instead of setting the source voltage of the floating gate cell as in Fig. 4, the implementation in [23] uses an input stage to drive the control gate voltage, which modulates the output current. Therefore, due to large gate capacitance and lower control gate referred transconductance, the implementation has a higher power–delay product when compared to the proposed architecture.

A translinear squaring circuit M7-M10 implements the kernel as shown in Fig. 7. Assuming that the transistors $M7, M8, M9$ and $M10$ are perfectly matched, the output current through $M10$ is given by

$$I_{\mathrm{out}} = I_{\mathrm{in}}^2/I_{sq} \tag{9}$$

where I_{sq} is the bias current through transistor $M11$. Even though under nonideal conditions the kernel computation will deviate from its perfect response, the requirement on the accuracy of nonlinearity is not stringent and is compensated by adapting the SVM training algorithm [17]. Fig. 8 shows measured response of the squaring circuit, which shows that the output current proportional to the square of the input current. The current then feeds to the second MVM comprising of MAC cell shown in Fig. 4 to produce SVM confidence functions corresponding to the general form (1). With $x_j, j = 1, \ldots, D$

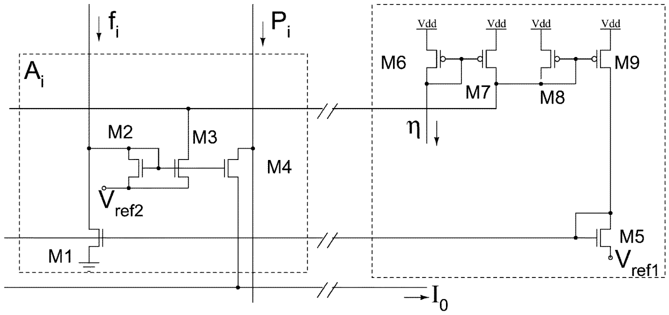


Fig. 9. Subtractive normalization circuit.

the components of the feature vector $\mathbf{x}[n]$ presented as input currents to MVM1, the current outputs $f_i, i = 1, \ldots, D$ of the chip are given by

$$f_i = \sum_{s=1}^{S} \lambda_{si} \frac{1}{I_{sq_s}} \left(\sum_{j=1}^{D} x_{sj} x_j \right)^2 . \tag{10}$$

The current bias I_{sq_s} of the squaring circuit is individually tunable to compensate for mismatch during calibration as described in Section V.

B. Normalization Circuit

The subtractive normalization according to (3) is implemented by the circuit shown in Fig. 9. The core, repeated for each category, is the basic building block A_i comprising M1-M4 which compute $[f_i(\mathbf{x}[n]) - Z[n]]_+$. The feedback circuit consisting of transistors M5-M9 determines the equilibrium condition in (3). If all the transistors are biased in weak-inversion, the network in Fig. 9 implements

$$\sum_{i}^{M} [f_i(\mathbf{x}[n]) - Z[n]]_+ = \eta + Z[n] e^{-V_{\text{ref}}/U_T}. \tag{11}$$

For $V_{\text{ref}} > 100$ mV, the circuit (11) implements the reverse water-filling equation (3). Transistor M4 scales the currents with the normalizing factor η and is used for off-chip current measurement during calibration and programming (Section V). Figs. 10(a) and (b) show the measured response of a four-class subtractive normalization circuit. The figures show the output current through transistor M4 corresponding to three classes (P2, P3, and P4) as a function of the output current of one of the classes (P1). For large value of the normalization factor η the change in the output response $P_i[n]$ is similar for all classes as predicted by the reverse water-filling criterion. Similarly for low value of η, the network demonstrates a saturating piece-wise linear response.

IV. Energy Efficiency, Noise, and Precision

In this section we analyze the energy efficiency and noise performance of the MAC cell in Fig. 4 and the normalization circuit in Fig. 9.

A. MAC Circuit

To achieve low power dissipation the MAC cell has to operate with small currents and yet achieve a desired computational bandwidth. Factors which limit the performance of the MAC cell are:

- minimum current required to achieve a specified computational bandwidth;
- minimum current required to maintain a specified signal-to-noise ratio (SNR).

The power dissipation of the MAC cell in Fig. 4 can be computed using the drain current I_{ds} through the transistor M4 that supplies current to all cells connected to node A. Let the supply voltage be denoted by V_{dd} then power dissipation of a single MAC cell is given by $P_d = 5 * I_{ds} V_{dd}$, where it is assumed that the source to drain voltage $V_{sd} \approx V_{dd}$. The factor 5 accounts for the bias current through transistor M3 which implies more current through transistor M4 than needed to supply the cell currents. Let the intrinsic capacitance of the output node of the cell be given by C_{cell} and the source referred transconductance of M4 be given by g_m. The power delay product for a single MAC operation determined by limitations on computational bandwidth is given by

$$P_d \tau \geq 5 * I_{ds} V_{dd} C_{\text{cell}} / g_m$$

or

$$P_d \tau \geq 5 * C_{\text{cell}} U_T V_{dd}. \tag{12}$$

where $g_m = I_{ds}/U_T$ in weak inversion. U_T is the thermal voltage which is ≈26 mV at room temperature. For a nominal value of $C_{\text{cell}} = 20$ fF and $V_{dd} = 4$ V, the minimum power–delay product due to the computational bandwidth limitation is given by $P_d \tau \approx 10^{-14}$ J/MAC.

The limit imposed by noise on power dissipation of the MAC is computed by accounting for thermal and flicker noise from the MOS transistors in the MAC cell. The noise power spectral density of the output current in Fig. 4 is given by [29], [30]

$$S_{I_d} = 2 \left(4 \gamma k T g_m + \frac{K_F I_{ds}^{A_F}}{f^{E_F} C_{ox} W L} \right) \tag{13}$$

where the factor 2 accounts for contribution from transistors $M1$ and $M2$. The high loop gain due to $M3$ suppresses the thermal noise contribution due to $M4$ and $M3$. The thermal noise parameter γ is approximately equal to $2/3$ in strong inversion. However in the subthreshold region, γ equals $1/(2\kappa)$ [29] and transconductance is proportional to drain current, $g_m = I_{ds}\kappa/(kT/q)$, reducing the thermal noise component to $2qI_{ds}$ per transistor, equivalent to two-sided (source/drain) shot noise [30]. Typical values of the flicker noise parameters K_F, A_F, and E_F in a 0.5μm CMOS process are $K_F = 2 \times 10^{-25}$, $A_F = 2$, and $E_F = 1$. Because of the square dependence on drain current, the relative contribution of flicker noise diminishes with diminishing bias currents, and is negligible in subthreshold. At 1 pA bias current, the flicker noise corner reaches 3×10^{-5} Hz.

To maintain a minimum level of signal to noise power ratio SNR for a bandwidth of $\Delta f Hz$, the drain current through the MAC cell has to satisfy

$$I_{ds} \geq \sqrt{4 q I_{ds} \Delta f \text{SNR}}$$

or

$$I_{ds} \geq 4 q \Delta f \text{SNR} \tag{14}$$

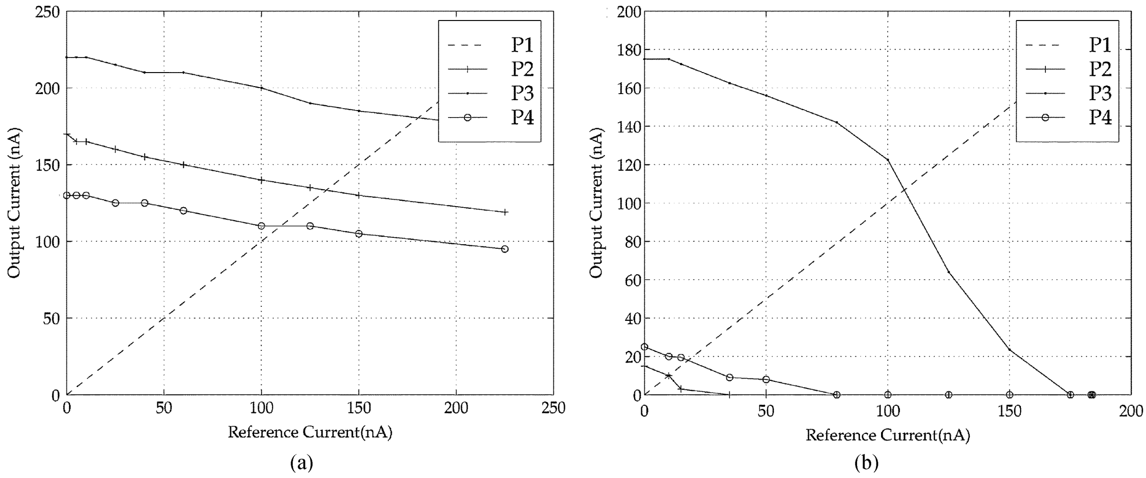


Fig. 10. Output current values P_i for different elements of the normalization network f_i corresponding to (a) large and (b) nominal value of the normalization constant η.

limiting the power–delay product to

$$P_d\tau \geq 4qV_{dd}\text{SNR}. \tag{15}$$

Therefore, to maintain at least unity (0 dB) SNR, the power–delay product of a single MAC cell is bounded from below by the largest of the two limits,

$$P_d\tau \geq \max(5C_{\text{cell}}U_T V_{dd}, 4qV_{dd}). \tag{16}$$

With $C_{\text{cell}} = 20$ fF, and $V_{dd} = 4$ V, the second term dominates the bound which remains $P_d\tau \geq 10^{-14}$. The bound is achieved for a bias current I_{ds} at least 75 fA. At this level the signal to noise power ratio SNR reaches 800, or 28 dB.

These values imply that for the MAC cell the power–delay product is dominated by the computational efficiency, rather than intrinsic noise of the analog computing array. The analysis ignored the effect of substrate coupling because the design uses continuous-time circuits which avoid digital switching. It also ignored diode leakage which at 75 fA drain current and at room temperature contributes significantly. For the 28,814 MAC cell array, the minimum theoretical power dissipation for a bandwidth of 80 Hz is 24 nW. Even lower limits are in principle feasible in deeper submicron processes. Note that unlike digital circuits in low-voltage deep submicron technology [31], subthreshold currents are not considered leakage but carry the signal and hence do not imply a lower limit on power dissipation.

B. Nonvolatile Storage

The precision of computation by the array is limited by the resolution of the MAC operation which in turn is limited by the precision of floating gate storage. Precision in floating gate programming is determined by programming duration and the accuracy of the read-out instrumentation [32]. In this paper a fixed current method, proposed in [33] has been used for programming the floating gate array. The method facilitates increased programming speed while achieving a precision of at least 7 bits, which in most cases is sufficient for recognition tasks. Fig. 11 shows a plot of 64 different floating gate cells programmed at an increment of 10 pA using the fixed current technique [33].

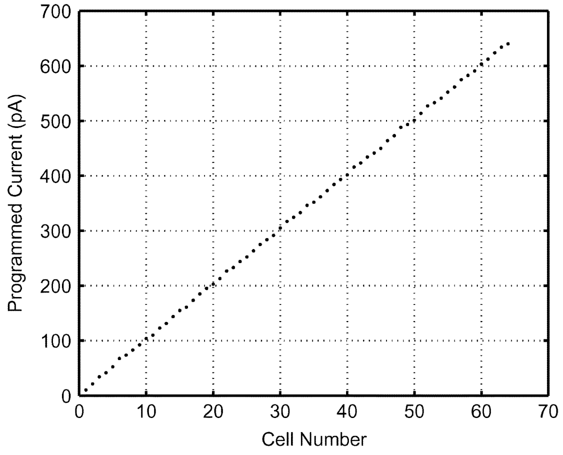


Fig. 11. Currents programmed on 64 different floating gate cells using the fixed current algorithm in [33] at an increment of 10 pA.

C. Normalization Circuit

Next we analyze the effect of thermal noise on the output of the normalization circuit in Fig. 9. For P active blocks satisfying the condition $[f_i(\mathbf{x}[n]) - Z[n]] > 0$ the contribution due to noise at the output transistor M3 is given by

$$I_P^2 \approx (4kT\gamma g_{m2} + 4kT\gamma g_{m3} + 4kT\gamma g_{m7}/P + 4kT\gamma g_{m1})\Delta f \tag{17}$$

where g_{mk} refers to transconductance of transistor M_k and γ is the transistor parameter defined in (13). According to (11), the noise contribution due to transistors M8, M9 and M5 can be ignored because of the high loop gain of the network and equals e^{V_{ref1}/U_T}. According to (17), the noise contribution due to the reference current transistor M7 can be reduced by increasing the size of the network (which increases the number of active blocks P). It can be seen that since the contribution of the thermal noise is additive, subtractive normalization is well suited as it reduces

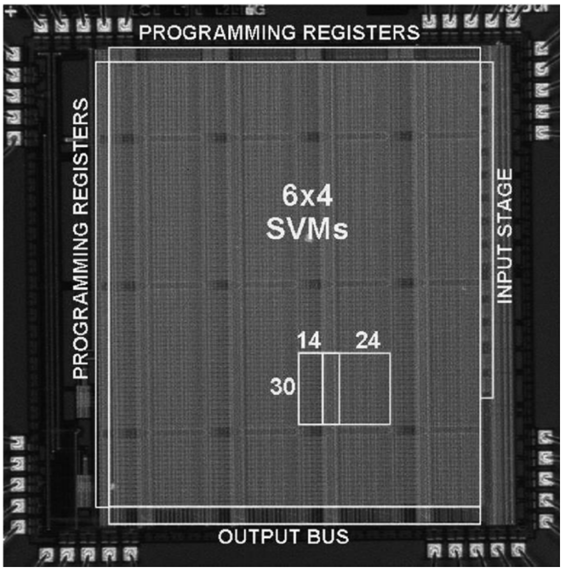


Fig. 12. Micrograph of the 14-input, 720-template, 24-class normalizing SVM classifier.

TABLE I
SVM CHIP CHARACTERISTICS

Technology	Value
Area	3mm×3mm
Technology	0.5μ CMOS
Supply Voltage	4 V
System Parameters	
Floating Cell Count	28814
Number of Support Vectors	720
Input Dimension	14
Number of States	24
Power Consumption	80nW - 840nW
Energy Efficiency	1.3 10^{12}MAC/s/W
@ 40 classification/s	

the contribution of noise from neighboring circuits A_i. Therefore, according to the value of normalization factor η, the minimum threshold level Z can be adaptively adjusted to match the noise level of the system.

V. Prototype Implementation and Calibration

A 14-input, 24-class, and 720-template SVM including a 24 class normalization network was implemented on a 3 mm × 3 mm chip, fabricated in a 0.5 μm CMOS process. Fig. 12 shows the micrograph of the fabricated chip and Table I summarizes its measured characteristics. An array of 6×4 SVMs, whose outputs are combined through an output bus, was implemented. The chip includes programming shift registers for selecting rows of floating gate cells. Any unused rows can be shut off during run-time, thus saving power.

All MAC cells in MVM array are randomly accessible for read and write operations. The programming registers are also used for calibration of the SVM chip. The calibration procedure compensates for mismatch between input and output paths by adapting the floating gate elements in the MVM cells.

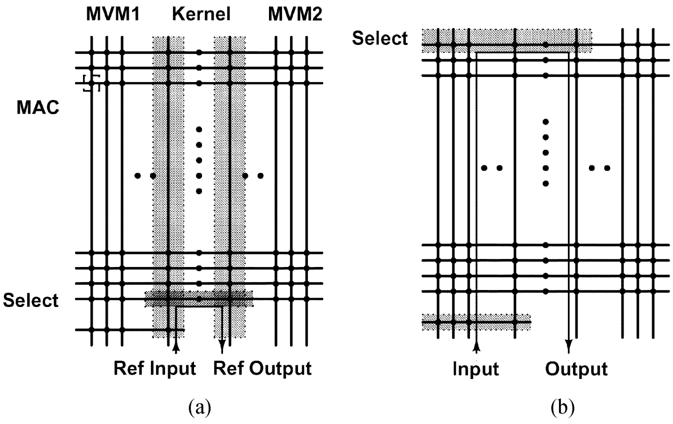


Fig. 13. Calibration procedure for the classifier chip. Compensation of mismatch due to (a) the squaring stages and (b) the input stages.

MOS transistors operating in weak inversion exhibit a greater mismatch than in strong inversion [34]. This can be attributed to a high transconductance to bias current ratio g_m/I_{ds} in the effect of threshold variations. Fortunately for transistors biased in weak-inversion, the dominant source of mismatch appears as a multiplicative factor because g_m/I_{ds} remains fairly constant for a wide range of currents I_{ds}. Following analysis of the signal path in Fig. 7, the cumulative effect of transistor mismatch on the output current (10) takes the form of input-referred multiplicative gain errors in the input current x_j, the square kernel bias current I_{sq} and the output current f_i. Fig. 14(a) shows measured currents at the output of the classifier when all the floating gate cells are programmed to an equal value and the input current is varied. The spread between the curves in Fig. 14(a) shows the degree of mismatch and also its multiplicative nature. The mismatch due to gate coupling parameter κ in (8), which affects the slope of the curve in Fig. 14(a), was measured to be less than 1% and hence was not considered in calibration.

The calibration circuits on the fabricated prototype consist of row and column scan shift registers which select individual rows and column of the floating gate array. Fig. 12(a) illustrates the selection procedure for automatic calibration. The intersection between each vertical and horizontal row denotes the location of a floating gate transistor whose value can be adapted during calibration. The key calibration steps are summarized as follows.

1) All the floating gate cells are programmed to a fixed current (10 nA for our experiments) using the fixed-current method described in [33].
2) The mismatch due to the I_{sq} in (10) is compensated by selecting a reference input column and a reference output column as shown in Fig. 13(a). Each row of the floating gate array is selected and the floating gate cell at the input stage of the coefficient MVM is adapted such that the outputs of the kernel are equal.
3) The mismatch due to the input stage is compensated for by selecting a reference row as shown in Fig. 13(b). Each input column is selected and the corresponding output is measured. The floating gate cell at the input stage of the support vector MVM is adapted such that the output current is set to a reference value.

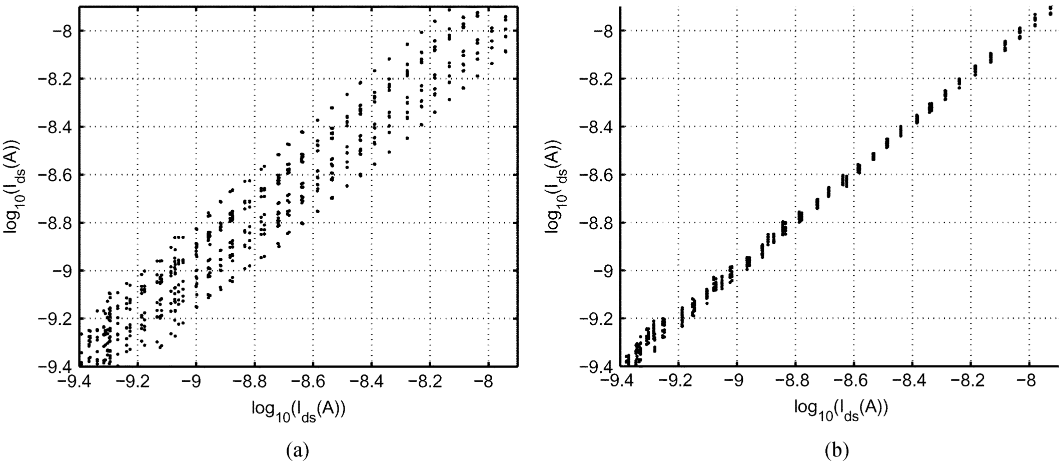


Fig. 14. Measured kernel current for different SVM stages (a) before calibration and (b) after calibration.

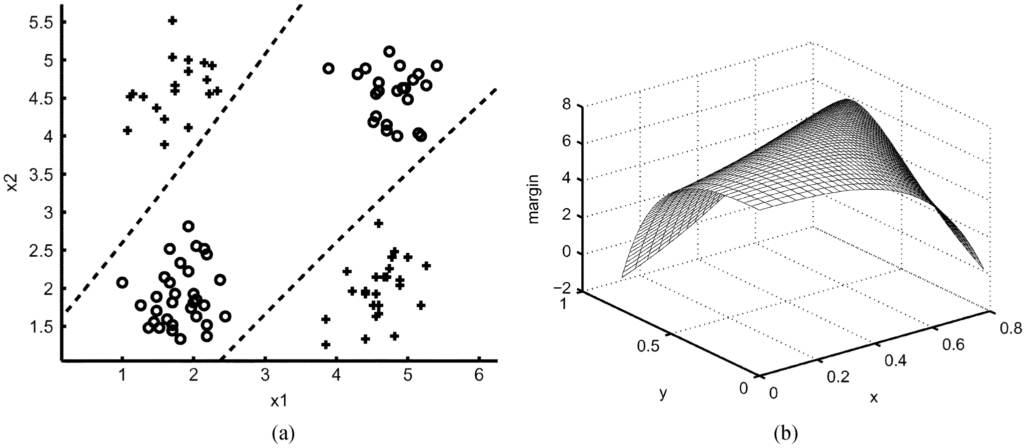


Fig. 15. (a) Decision boundary of an example two dimensional, binary class classification problem trained using a square kernel. (b) Classification manifold showing the variation of the decision function for different values of inputs.

4) The support vectors and the coefficient vectors are programmed onto the floating gate array with respect to the reference current.

Fig. 14(a)–(b) demonstrates the effect of calibration, showing the kernel output when different rows are selected and the input current is varied. After calibrating the kernel stage using coefficient stage parameters, the kernel response plot shows a reduced spread and hence the procedure alleviates the effect of mismatch.

VI. Experiments and Results

An experimental setup was designed to evaluate the performance of the classifier chip. A custom circuit board consisting of a bank of digital-to-analog converters was developed to generate acoustic features for the SVM chip. A lookup table, calibrated before and after each experiment, maps the input current vector into voltage vectors generated by the DAC, avoiding bandwidth limitations in sourcing very small currents into the chip. An external *I–V* converter measures the output current from the chip. For practical use, the presented design is intended for a system-on-chip application where the classifier is embedded with a front-end processor and quantizer. An on-chip integrated decision circuit performs two-level conversion of the output current to produce SVM outputs.

The first set of experiments used a simulated classification task to benchmark the performance of the SVM and characterize its power dissipation. Fig. 15 plots the feature vectors on which the SVM was trained. The symbols “o” and “x” indicate binary class membership of each feature vector. The decision boundary separating the two classes is shown by a dashed line. A GiniSVM toolkit [35] was used to train the SVM classifier and the subtractive normalization in (3) was naturally embedded in the training algorithm. The classifier generated two confidence functions corresponding to each of the binary classes. Fig. 15(b) plots the difference of the two SVM confidence functions (denoted by margin) versus the two dimension feature vector. The value of the margin relative to a threshold determines class membership of any two dimensional vector. The parameters obtained through SVM training were programmed onto a calibrated chip. The input feature vectors were then presented to the chip using the DAC array. A plot of margin values, obtained by measuring the output current from the chip, are shown in Fig. 16(a)–(b). The power dissipation of the chip was adjusted by changing the internal bias conditions. The figure

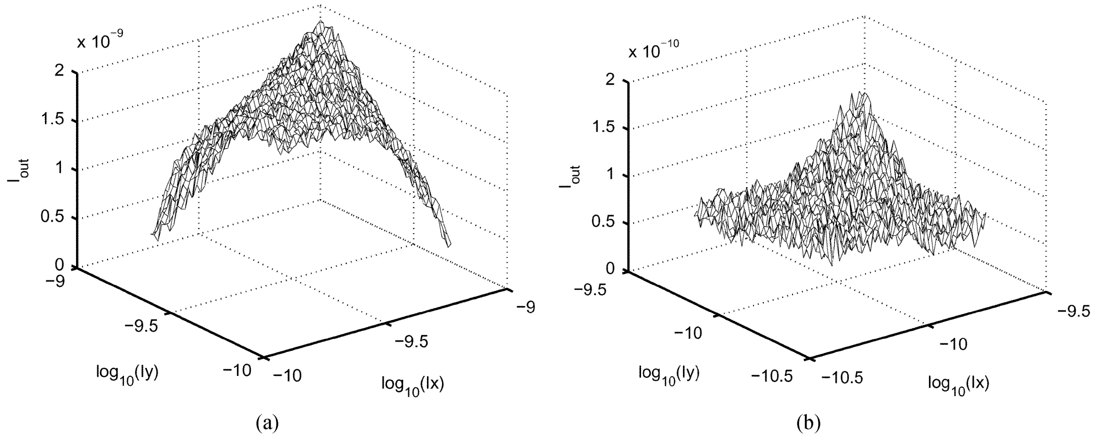


Fig. 16. Measured classification manifold for power dissipation levels at (a) 5 μW and (b) 500 nW.

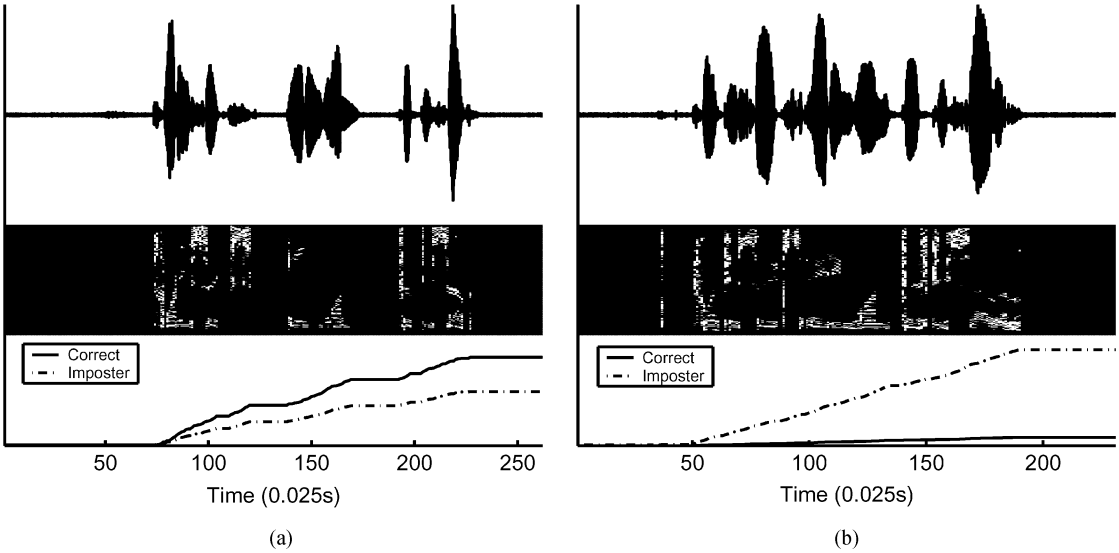


Fig. 17. Speaker verification with the normalizing SVM classifier chip. (a) Verification of a correct speaker, whose confidence values is integrated over time. (b) Rejection of an imposter speaker.

shows that the response of the classifier is similar to the simulated response when the bias currents are set to 5 μW of total power consumption. Even at 500 nW of total power consumption, the decision boundary is consistent although noise and limited accuracy of the external measurement instrumentation contaminate the shape of the measured decision manifold. The on-chip decision circuits producing SVM outputs avoid the instrumentation errors in measuring very small off-chip currents [34].

For the second set of experiments the SVM chip was programmed to perform speaker verification using speech data from YOHO corpus. 480 utterances corresponding to 10 separate speakers (speaker IDs: 101–110) were chosen. For each of these utterances, 12 dimensional mel-cepstra coefficients were computed for every 25 ms speech frame. These coefficients were combined using k-means clustering to obtain 50 clusters per speaker, which were then used for training the classifier. For testing 480 utterances for those speakers were chosen, and confidence scores returned by the classifier were integrated over all frames of an utterance according to (4). A sample verification result, as measured using the SVM chip is shown in Fig. 17(a)–(b) where the true speaker is identified and an imposter is rejected. A receiver operating characteristic (ROC) curve was computed over 480 out-of-sample test utterances to evaluate the generalization performance of the classification chip. The curve plots the number of imposter speakers versus number of correct speakers verified by the system for different threshold parameter in (5). Fig. 18 compares ROC curves obtained through measurement with those obtained through simulations. The verification system demonstrates 97% true acceptance at 3% false positive rate, similar to the performance obtained through floating point software simulations. The power consumption of the SVM chip for speaker verification task is only 840 nW, demonstrating its suitability for autonomous sensor applications. At 40 classifications per second (80 Hz analog bandwidth), the chip attains an energy efficiency of $1.3 \times 10^{1}2$ MACs per Joule.

VII. Discussion and Conclusion

Ultra-low power smart sensors require power efficient recognition systems to identify patterns of interest in their environment. We showed an example where analog VLSI provides

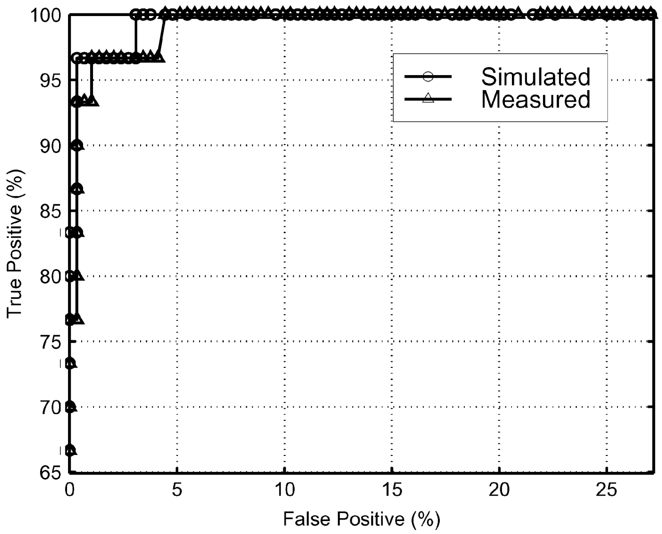


Fig. 18. Measured and simulated ROC curve for the speaker verification experiment.

an attractive alternative to digital signal processing systems for implementing ultra-low-power sensors. This paper presented a sub-microwatt analog VLSI classifier for acoustic signature identification, and other applications of signal detection for RF-ID biometrics or implantable biomedical monitoring. The architecture uses an array of floating gate elements for nonvolatile storage and computation. Nonvolatile storage of parameters makes the system suitable for sensors powered by energy harvesting techniques where power disruption is frequent. All analog processing on the chip is performed by transistors operating in weak-inversion resulting in power dissipation in levels ranging from nanowatts to microwatts. Compensation of analog imperfections due to mismatch and nonlinearity is performed through a systematic calibration and PC-in-loop training procedure. A prototype implementing the proposed architecture has been fabricated in a 0.5 μm CMOS process and has been demonstrated on a speaker verification task at sub-microwatt power.

References

[1] A. Wang and A. P. Chandrakasan, "Energy-efficient DSPs for wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 19, no. 4, pp. 68–78, Jul. 2002.

[2] S. Meninger, J. O. Mur-Miranda, R. Amirtharajah, A. Chandrakasan, and J. H. Lang, "Vibration-to-electric energy conversion," *IEEE Trans. Very Large Scale Integration (VLSI)*, vol. 9, no. 2, pp. 64–76, Feb. 2001.

[3] J. Rabaey, J. Ammer, T. Karalar, S. Li, B. Otis, M. Sheets, and T. Tuan, "PicoRadios for wireless sensor networks: the next challenge in ultra-low power design," in *IEEE ISSCC 2002 Dig. Tech. Papers*, San Francisco, CA, Feb. 2002, pp. 200–201.

[4] J. F. Randall, "On ambient energy sources for powering indoor electronic devices," Ph.D. dissertation, Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland, May 2003.

[5] R. Amirtharajah and A. P. Chandrakasan, "A micropower programmable DSP using approximate signal processing based on distributed arithmetic," *IEEE J. Solid-State Circuits*, vol. 39, no. 2, pp. 337–347, Feb. 2004.

[6] E. A. Vittoz, "Low-power design: Ways to approach the limits," in *IEEE ISSCC 1994 Dig. Tech. Papers*, San Francisco, CA, 1994, pp. 14–18.

[7] A. G. Andreou, "On physical models of neural computation and their analog VLSI implementation," in *Proc. Workshop on Physics and Computation*, Nov. 1994, pp. 255–264.

[8] P. Leong and M. Jabri, " A low-power VLSI arrhythmia classifier," *IEEE Trans. Neural Networks*, vol. 6, no. 6, pp. 1435–1445, Nov. 1995.

[9] T. Yamasaki, K. Yamamoto, and T. Shibata, "Analog pattern classifier with flexible matching circuitry based on principal-axis-projection vector representation," in *Proc. 27th Eur. Solid-State Circuits Conf. (ESSCIRC 2001)*, Sep. 2001, pp. 197–200.

[10] M. S. Shakiba, D. A. Johns, and K. W. Martin, "BiCMOS circuits for analog Viterbi decoders," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 45, no. 12, pp. 1527–1537, Dec. 1998.

[11] J. Lazzaro, J. Wawrzynek, and R. P. Lippmann, "A micropower analog circuit implementation of hidden Markov model state decoding," *IEEE J. Solid-State Circuits*, vol. 32, no. 8, pp. 1200–1209, Aug. 1997.

[12] P. Kinget and M. Steyaert, "Analog VLSI design constraints of programmable cellular neural networks," *Analog Integr. Circuits Signal Process.*, vol. 15, no. 3, pp. 251–261, Mar. 1998.

[13] B. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifier," in *Proc. 5th Annu. ACM Workshop on Computational Learning Theory*, 1992, pp. 144–152.

[14] V. Vapnik*, The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[15] R. Genov and G. Cauwenberghs, "Charge-mode parallel architecture for vector-matrix multiplication," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 48, no. 10, pp. 930–936, OCt. 2001.

[16] S. Chakrabartty and G. Cauwenberghs, "Sub-microwatt analog VLSI support vector machine for pattern classification and sequence estimation," in *Proc. Neural Information Processing Systems Conf. (NIPS'2004)*. Cambridge, MA: MIT Press, 2005.

[17] S. Chakrabartty and G. Cauwenberghs, "Forward decoding kernel machines: A hybrid HMM/SVM approach to sequence recognition," in *Proc. SVM 2002*, pp. 278–292, Lecture Notes in Computer Science, 2388.

[18] S. Chakrabartty and G. Cauwenberghs, "Margin normalization and propagation in analog VLSI," in *Proc. IEEE ISCAS 2004*, Vancouver, Canada, 2004, pp. I-901–904.

[19] H.-A. Loeliger, "Probability propagation and decoding in analog VLSI," in *Proc. IEEE Int. Symp. Information Theory*, Cambridge, MA, 1998, p. 146.

[20] F. Lustenberger, "An analog VLSI decoding technique for digital codes," in *Proc. IEEE ISCAS'99*, 1999, vol. 2, pp. 424–427.

[21] T. A. Cover and J. A. Thomas*, Elements of Information Theory*. New York: Wiley, 1991.

[22] A. Kramer, "Array-based analog computation," *IEEE Micro*, vol. 16, no. 5, pp. 40–49, May 1996.

[23] A. Aslam-Siddiqi, W. Brockherde, and B. Hosticka, "A 16 × 16 nonvolatile programmable analog vector-matrix multiplier," *IEEE J. Solid-State Circuits*, vol. 31, no. 10, pp. 1502–1509, 1998.

[24] T. Serrano-Gotarredona, B. Linares-Barranco, and A. G. Andreou, "A general translinear principle for subthreshold MOS transistors," *IEEE Trans. Circuits Syst., I: Fundam. Theory Applicat.*, vol. 46, no. 5, pp. 607–616, May 1999.

[25] B. Gilbert, "Translinear circuits: A proposed classification," *Electron. Lett.*, vol. 11, no. 1, pp. 14–16, Jan. 1975.

[26] C. Diorio, P. Hasler, B. Minch, and C. A. Mead, "A single-transistor silicon synapse," *IEEE Trans. Electron Devices*, vol. 43, no. 11, pp. 1972–1980, Nov. 1996.

[27] A. Andreou and K. Boahen, "Translinear circuits in subthreshold MOS," *J. Analog Integrated Circuits Signal Process.*, vol. 9, no. 2, pp. 141–166, Mar. 1996.

[28] T. Shibata and T. Ohmi, "A functional MOS transistor featuring gate-level weighted sum and threshold operations," *IEEE Trans. Electron Devices*, vol. 39, no. 6, pp. 1444–1455, Jun. 1992.

[29] Y. P. Tsividis*, Operation and Modeling of the MOS Transistor*. New York: McGraw-Hill, 1988.

[30] R. Sarpeshkar, T. Delbruck, and C. A. Mead, "White noise in MOS transistors and resistors," *IEEE Circuits Devices Mag.*, vol. 9, no. 6, pp. 23–29, Nov. 1993.

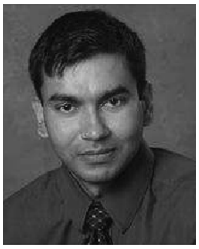
[31] B. H. Calhoun, A. Wang, and A. Chandrakasan, "Modeling and sizing for minimum energy operation in subthreshold circuits," *IEEE J. Solid-State Circuits*, vol. 40, no. 9, pp. 1778–1786, Sep. 2005.

[32] A. Bandyopadhyay, G. J. Serrano, and P. Hasler, "Programming analog computational memory elements to 0.2% accuracy over 3.5 decades using a predictive method," in *Proc. IEEE ISCAS 2005*, pp. 2148–2151.

[33] S. Chakrabartty and G. Cauwenberghs, "Fixed current method for programming large floating gate arrays," in *IEEE ISCAS 2005*, pp. 3934–3937.

[34] B. Linares-Barranco and T. Serrano-Gotarredona, "On the design and characterization of femtoampere current-mode circuits," *IEEE J. Solid State Circuits*, vol. 38, no. 8, pp. 1353–1363, Aug. 2003.

[35] Gini-SVM Toolkit. [Online]. Available: http://bach.ece.jhu.edu/svm/ginisvm



Shantanu Chakrabartty (M'96) received the B.Tech. degree from the Indian Institute of Technology, Delhi, India, in 1996, the M.S. and Ph.D. degrees in electrical engineering from Johns Hopkins University, Baltimore, MD, in 2001 and 2004, respectively.

He is currently an Assistant Professor in the Department of Electrical And Computer Engineering, Michigan State University. From 1996 to 1999, he was with Qualcomm Inc,, San Diego, CA, and during 2002 he was a visiting researcher at the University of Tokyo. His current research interests include low-power analog and digital VLSI systems, hardware implementation of machine learning algorithms with application to biosensors and biomedical instrumentation.

Dr. Chakrabartty was a recipient of The Catalyst foundation fellowship from 1999 to 2004 and won the best undergraduate thesis award in 1996. He is currently a member for IEEE BioCAS technical committee and IEEE Circuits and Systems Sensors technical committee.

Gert Cauwenberghs (SM'89–M'94–S'04) received the Ph.D. degree in electrical engineering from the California Institute of Technology, Pasadena, in 1994.

He was previously Professor of electrical and computer engineering at Johns Hopkins University, Baltimore, MD. He joined the University of California at San Diego, La Jolla, as Professor of neurobiology in 2005. His research aims at advancing silicon adaptive microsystems to understanding of biological neural systems, and to development of sensory and neural prostheses and brain-machine interfaces.

Dr. Cauwenberghs received the NSF Career Award in 1997, ONR Young Investigator Award in 1999, and Presidential Early Career Award for Scientists and Engineers in 2000. He is Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I, IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING, and *IEEE Sensors Journal*.