Realizing Low-Energy Classification Systems by Implementing Matrix Multiplication Directly Within an ADC

Zhuo Wang, *Student Member, IEEE*, Jintao Zhang, *Student Member, IEEE*, and Naveen Verma, *Member, IEEE*

***Abstract*—In wearable and implantable medical-sensor applications, low-energy classification systems are of importance for deriving high-quality inferences locally within the device. Given that sensor instrumentation is typically followed by A-D conversion, this paper presents a system implementation wherein the majority of the computations required for classification are implemented within the ADC. To achieve this, first an algorithmic formulation is presented that combines linear feature extraction and classification into a single matrix transformation. Second, a matrix-multiplying ADC (MMADC) is presented that enables multiplication between an analog input sample and a digital multiplier, with negligible additional energy beyond that required for A-D conversion. Two systems mapped to the MMADC are demonstrated: (1) an ECG-based cardiac arrhythmia detector; and (2) an image-pixel-based facial gender detector. The RMS error over all multiplication performed, normalized to the RMS of ideal multiplication results is 0.018. Further, compared to idealized versions of conventional systems, the energy savings obtained are estimated to be 13× and 29×, respectively, while achieving similar level of performance.**

***Index Terms*—ADC, boosting, classification, embedded sensing.**

I. Introduction

MEDICAL sensors used for long-term patient monitoring, especially outside the hospital setting, offer the potential to substantially improve patient health, quality of life, and outcomes [1], [2]. This potential depends on two aspects: (1) the ability to acquire signals that are informative with respect to a patient state; and (2) the ability to make relevant inferences from such signals. Systems to address this face a wide range of technical challenges, especially when size and energy constraints, necessary for long-term patient comfort and compliance, are considered. Motivated by recent progress in wearable and implantable sensor technologies (e.g., [3]), which are making a wide range of patient signals accessible within such systems, this paper focuses on deriving inferences at minimal energy levels. Generally speaking, A-D conversion follows sensor instrumentation, and digital computation follows A-D conversion to derive the inferences. This paper presents an ADC that performs data conversion as well as most of the computations required to derive the inferences within a single stage. In particular, this paper presents (1) an algorithmic formulation that combines the operations of linear feature extraction and classification into a single matrix transformation, and (2) an ADC architecture that performs multiplication between analog input samples and an arbitrary matrix of digital elements. The ADC architecture is referred to as a matrix-multiplying ADC (MMADC) [4].

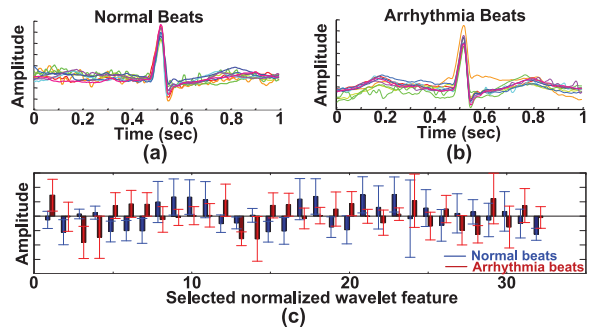


Fig. 1. ECG patient data, showing (a) normal beats, (b) arrhythmia beats, and (c) the discrete wavelet transform of both types of beats, as biomarkers for detecting cardiac arrhythmia.

The focus for deriving the inferences is on machine-learning algorithms. The benefit of machine-learning algorithms is motivated, on a high level, using the example considered in Fig. 1. Electrocardiogram (ECG) data from a patient is shown [5], [6], with beats in Fig. 1(a) being expert identified as normal beats and those in Fig. 1(b) being expert identified as arrhythmias (e.g., premature ventricular contraction, bundle branch block beats, atrial premature beats, etc. [5]). We see, especially from looking at the normal beats, that substantial physiologic variances are prominent. This makes it difficult to detect the pathophysiologic variances of the arrhythmias, which can be much more minute. For example, time-frequency analysis of the ECG is often used to expose biomarkers of arrhythmias [7]. Fig. 1(c) plots discrete-wavelet-transform (DWT) coefficients, for normal beats (blue) and arrhythmias (red), with average value and standard deviation shown over 2500 beats. While for each coefficient, differences between the two classes are noticeable, limited analytical understanding of such differences

Manuscript received July 30, 2015; revised October 27, 2015; accepted November 07, 2015. Date of publication December 28, 2015; date of current version January 27, 2016. This work was supported by SRC, NSF (CCF-1253670), and Systems on Nanoscale Information fabriCs (SONIC), one of the six SRC STARnet Centers, sponsored by MARCO and DARPA. This paper was recommended by Associate Editor R. Genov.

The authors are with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: zhuow@princeton.edu; jintao@princeton.edu; nverma@princeton.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBCAS.2015.2500101

and large variances within the classes make it difficult to establish a robust decision rule. Related to this, both the differences between the classes and the variances within each class are patient specific, making it likely that patient-customized decision rules can substantially improve accuracy [8], [9].

This points to methods that can model statistical correlations across a potentially large number of biomarkers (DWT coefficients) for enhanced robustness, and can do so efficiently on a patient-by-patient basis. Machine-learning algorithms enable data-driven methods for efficiently constructing decision rules by modeling the statistics over a potentially large number of dimensions. By starting with data, such modeling can be performed to high order, without the need for strong analytical representations. Thus, machine-learning algorithms are gaining importance for performing inferences (e.g., classification) across a broad range of physically-complex embedded signals, but especially physiological signals acquired from medical sensors [10]. With the increasing prominence of wearable and implantable sensor technologies, the ability to perform such inferences within a similar form-factor, namely under severe battery constraints, can open up important system and use-case possibilities [11].

The remaining paper is structured as follows. Section II presents the system and algorithmic formulation for reducing feature extraction and classification via a machine-learning algorithm to a single matrix transformation. Then, Section III presents an overview of the MMADC architecture, and Section IV presents the circuit design details. Finally, Section V presents testing results of the IC prototype, and Section VI presents application system demonstrations.

II. Overview of System Formulation

Fig. 2 shows the typical structure of a classification system. The input signal from a sensor $x_A(t)$ is first fed to an ADC, so that digital computations may subsequently be performed. On a high level, these computations correspond to feature extraction and classification. Feature extraction involves creating a representation of the signal that enhances pattern recognition. Most notably, this may imply dimensionality reduction and/or computation of specific biomarkers. Though the precise feature-extraction computations can be highly variable across applications, simple linear operations are commonly used; for instance, the system demonstrations presented in Section VI employ principal components analysis (PCA) for dimensionality reduction and discrete wavelet transform (DWT) for computing biomarkers. The resulting representation is referred to as a feature vector, designated as $\vec{u}$. Feature vectors can be viewed as points in a hyper-dimensional space, called a feature space. Classification then involves mapping feature vectors to logical classes. In typical machine-learning algorithms, this is achieved through two phases. The first phase is training, which occurs off-line and typically infrequently; hence, its energy is not the primary focus. Training uses previously observed instances of data represented in the feature space $\vec{u}_{train}$, together with their known class labels, to form a model, often in the form of a hyperplane which then serves as a decision boundary. The second phase is testing (detection), which occurs continuously and in real time; hence, its energy is a primary concern. Testing employs the model to make class decisions, designated numerically as d. Often a threshold is applied for mapping to binary classes, leading to d_{thresh}. The subsections below look more closely at the complexity of the feature-extraction and classification operations, and then lead to the algorithmic formulation proposed.

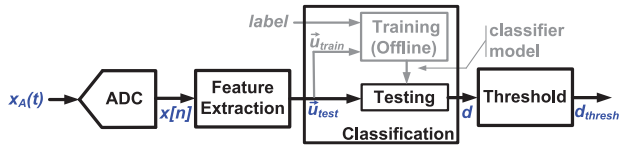


Fig. 2. System diagram of conventional classification system.

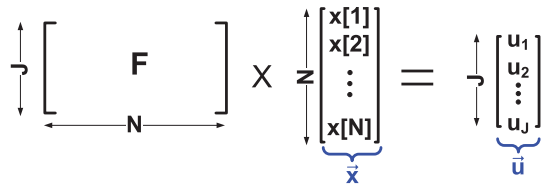


Fig. 3. Complexity of linear feature extraction in conventional system corresponds to roughly $J \times N$ multiplications and $J \times (N-1)$ additions.

A. System Complexity

Assuming linear feature extraction within the conventional system, the computations required can be represented as shown in Fig. 3. Namely, as multiplication of a matrix $\mathbf{F}$ with a vector $\vec{x}$, corresponding to N samples of the input $x[n]$. For a feature-vector dimensionality of J, this amounts to $J \times N$ multiplications and $J \times (N-1)$ additions.

For classification, we consider as an example a support-vector machine (SVM), which is commonly used to implement a *strong classifier*. A strong classifier is defined as one whose model can be fitted to arbitrary training data, while a *weak classifier* is defined as one whose model cannot. The computation required is shown in (1). Here, $\vec{v_i}$ correspond to S training feature vectors selected to a set called the support vectors in order to represent the decision boundary, while γ, b correspond to training parameters. Of key importance for realizing a strong classifier is the kernel function $K(\cdot)$, which introduces non-linearity to achieve flexibility in the decision boundary. The $K(\cdot)$ shown in (2) is for a radial-basis function (RBF) kernel

$$d = \sum_{i=1}^{S} K(\vec{u}, \vec{v}_i) - b \tag{1}$$

$$= \sum_{i=1}^{S} exp(\gamma \|\vec{u} - \vec{v}_i\|^2) - b. \tag{2}$$

Fig. 4 shows the importance of non-linear $K(\cdot)$ using feature-vector data derived from the ECG in an arrhythmia-detection application [7] (due to high dimensionality of the actual feature vectors, the data is projected to two dimensions using PCA for visualization). The figure also shows the decision boundaries that result from training a linear SVM and an RBF SVM. As seen, the linear decision boundary achieves inadequate fitting, resulting in misclassification of many points; generally,

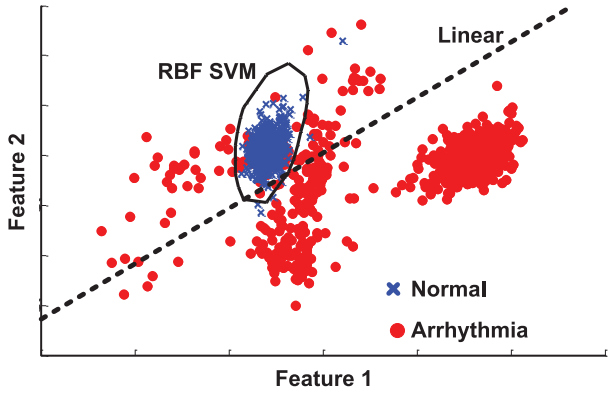


Fig. 4. Feature-vector data in arrhythmia detection application shown with linear and RBF SVM decision boundaries derived from training.

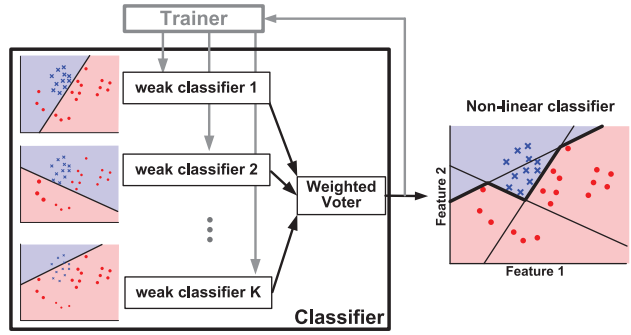


Fig. 5. Illustration of AdaBoost algorithm for creating a strong classifier based on multiple weak linear classifiers.

non-linear $K(\cdot)$ is necessary for adequate fitting [12]. Taking the computation in (1), approximately $J \cdot S$ multiplications and $2 \cdot J \cdot S$ additions are required. Additionally, S non-linear (exponential) operations are required, which can be implemented using a CORDIC.

B. Proposed Algorithmic Formulation

The proposed algorithmic formulation is based on combining feature-extraction computations with classification computations. Opportunities to achieve such combination have been previously explored and leveraged towards efficient system implementation. For instance, [13], [14] focuses on SVM classification using weaker kernels than the RBF, but wherein linear operations dominate. In this case, the total operations required scale with the dimensionality of $\vec{x}$ and the number of support vectors S. However, the proposed formulation focuses on linear classifiers. Though these lead to inadequate fitting, a method known as *boosting* is leveraged whereby fitting at the level of a strong classifier can be achieved using multiple linear classifiers, typically much fewer than the number of support vectors required in an SVM kernel.

The system computation required for *both* linear feature extraction and classification when a linear classifier is used is shown in (3). The linear hyperplane is defined by a $J \times 1$ classification vector $\vec{c}$. With the feature vector $\vec{u}$ is computed by multiplication between the $J \times N$ matrix $\mathbf{F}$ and the N-dimensional vector $\vec{x}$, the classification vector $\vec{c}$ can in fact be combined with the feature-extraction matrix $\mathbf{F}$ into a single $1 \times N$ matrix $\mathbf{H}$. This avoids explicit computation of the feature vector $\vec{u}$. Considering that the feature vector is an intermediate representation which provides more general information than required for classification, avoiding its computation has the potential to reduce system complexity. In particular, the entire classification system can now be reduced to just N multiplication operations and $N-1$ addition operations.

$$d = \vec{c}^{T} \times \vec{u} \tag{3}$$
$$= \vec{c}^{T} \times \mathbf{F} \times \vec{x} \tag{4}$$
$$= \mathbf{H} \times \vec{x} \tag{5}$$

However, the inadequate fitting of linear classifiers must be overcome. For this, we consider a machine-learning algorithm known as adaptive boosting (AdaBoost) [15]. AdaBoost uses an ensemble of weak classifiers to construct a strong classifier. The algorithm is illustrated in Fig. 5. Here, weak classifiers are trained iteratively, such that data instances misclassified in a given iteration are given increased weight during training of the subsequent iteration. The total number of iterations required, designated as K, is determined from training. Once the K weak-classifier decisions are obtained, these are fed to a weighted voter to arrive at the final classification decision. The voter weights are also determined during AdaBoost training, and the voter hardware simply amounts to a signed adder, adding or subtracting a weight depending on the decision of the corresponding weak classifier. As shown in Fig. 5, for the case of linear weak classifiers, the AdaBoost algorithm effectively results in a decision boundary formed from the multiple weak classifiers, thereby achieving fitting at the level of a strong classifier. In fact, theoretical work on AdaBoost has shown that perfect fitting to the training set can be achieved with very weak classifiers, whose performance is only marginally better than 50/50 guessing [15].

Compared to a conventional strong classifier, such as SVM with RBF kernel, AdaBoost raises a number of benefits for potentially reducing system complexity. First, the low performance requirement of the weak classifiers allows us to choose weak classifiers that are preferred from the perspective of implementation resources. For instance, this leads to straightforward mixed-signal implementation within the MMADC. Further, as described in the following subsection, it affords robustness against non-idealities in the weak-classifier implementations. Second, particularly when linear weak classifiers are used, it avoids explicit computation of the feature vector $\vec{u}$, which is necessary when non-linear classifier kernels are employed. As shown in Fig. 6, with K iterations, the $\vec{c_1} \ldots \vec{c_K}$ weak classification vectors can now be represented as a $K \times J$ matrix. As before, this matrix can now be combined with the feature-extraction matrix $\mathbf{F}$, reducing most of the feature extraction and classification computations to multiplication by a single $K \times N$ matrix $\mathbf{H}$. This requires $K \times N$ multiplications and $K \times (N-1)$ additions. After thresholding the elements of the resultant vector, the weak classifier decisions are obtained

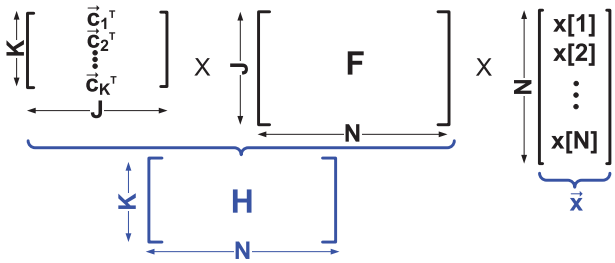


Fig. 6. Computational complexity of AdaBoost with Linear Classifiers [4].

TABLE I
ENERGY COMPARISON

Applications	Conventional System	Proposed System	Savings
ECG arrhythmia detection (N=256, J=256, S=170, K=5)	Mults: 109,056 Adds: 152,320	Mults: 1,280 Adds: 1,279	Mults: 85× Adds: 119×
Image-pixel gender detection (N=19200, J=200, S=180, K=1)	Mults: 3,876,000 Adds: 3,911,800	Mults: 19,200 Adds: 19,199	Mults: 202× Adds: 204×

and can be fed to a weighted voter (requiring $K-1$ further additions) to yield a final classification decision.

As we see from estimating the operations required, the proposed formulation introduces a complexity tradeoff between the number of weak classifier iterations K and the dimensionality of the feature vector J. Because the weak classifiers are directly optimized (through the AdaBoost algorithm) to achieve a strong classification result, K has the potential to be small; on the other hand, J is not directly optimized for classification. However, we point out that this will not always be the case, generally depending on the strength of the weak classifier, the complexity of the features, and the distribution of the application data. To be more concrete, we consider the system demonstrations presented in Section VI: (1) ECG-based cardiac arrhythmia detection; and (2) image-pixel-based gender detection. Table I compares the operations required for a conventional system based on RBF SVM (Fig. 2) with those for the proposed system. The proposed formulation leads to over 85× and over 200× reduction, respectively, in the operations.

C. Robustness for Mixed-Signal Implementation

As mentioned, the proposed algorithmic formulation is realized via a mixed-signal implementation within the MMADC. As detailed in the following section, such an implementation achieves low energy and low hardware complexity, but is also somewhat affected by circuit non-idealities. Previous work has shown how the AdaBoost algorithm can enable an approach whereby static random non-idealities in the implementation of the weak classifiers can be overcome during the training process. The approach is referred to as Error-Adaptive Classifier Boosting (EACB) [16], [17]. EACB leverages iterative training to not only overcome classifier fitting errors (as in the case of AdaBoost), but also classifier errors due to non-idealities in the implementation. The approach is illustrated in Fig. 7, where the ideal weak-classifier decision boundaries are shown as being perturbed due to such errors. By using the actual non-ideal weak-classifiers implemented, training data weights are assigned such that subsequent iterations are specifically trained to correct errors from the previous iterations. Through

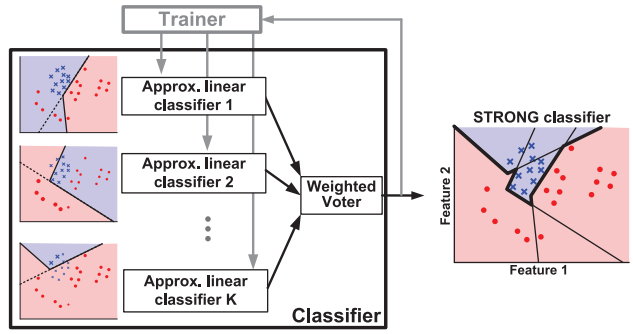


Fig. 7. Illustration of Error-Adaptive Classifier Boosting (EACB) approach, to correct errors due to non-ideal implementation of weak classifiers, in addition to fitting errors [17].

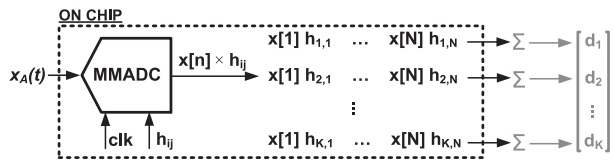


Fig. 8. MMADC conversion process for matrix multiplication.

such a data-driven approach, adaptation to overcome complex systematic non-idealities can be achieved without additional hardware, simply by applying the training data to one or a small number of implementations (to represent the systematic component of non-idealities). Generally, chip-to-chip variable component of non-idealities can also be handled, but by applying the training data to each implementation.

III. OVERVIEW OF MATRIX MULTIPLYING ADC (MMADC)

Analog implementation of classifiers has been explored in various contexts. For instance, [13] presents a classifier taking digital inputs but leveraging analog charge coupling to implement approximate multiply-accumulate operations, while [18] presents a classifier based on thin-film transistors to reduce the data from thin-film sensors. In the presented MMADC, the aim is to expand the A-D conversion operations performed in conventional systems (Fig. 2) to include multiplication at nearly no additional energy and throughput cost. The MMADC presents an architecture wherein the conversions can thus be organized to implement matrix multiplication between input samples, which represent a vector, and stored digital samples, which represent an arbitrary matrix. Taking the algorithmic formulation from above, which combines most of the feature extraction and classification computations into multiplication by a single $K \times N$ matrix $\mathbf{H}$, the MMADC substantially mitigates the need for additional computations following A-D conversion in a classification system.

Fig. 8 illustrates the MMADC conversion process, involving two inputs: (1) an analog sample $x_A(nT)$, corresponding to the input signal; and (2) a digital multiplier $h_{i,j}$, corresponding to the entries of the $\mathbf{H}$ matrix. In matrix multiplication, each analog sample is converted and applied a different digital multiplier corresponding to each of the K rows. Recalling that K refers to the number of weak-classifier iterations, in cases involving a

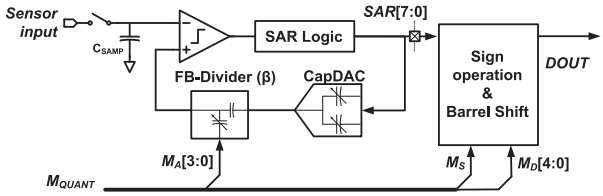


Fig. 9. Hardware block diagram of the MMADC [4].

small number of iterations, the MMADC can lead to system energy that is on the order of just an ADC. Following the multiplication operations, matrix multiplication is completed by storing the outputs and finally adding together those from each row to yield the elements of the resultant vector. In the implementation presented, this addition is performed off chip.

Fig. 9 shows a detailed hardware block diagram of the MMADC [4], which is based on a SAR architecture. Multiplication requires analog-domain computation, which raises two major challenges. First, active analog circuits typically suffer from substantial non-idealities, which makes their operation imprecise or requires considerable energy to overcome. Second, multiplication substantially increases the dynamic range requirements, and, in the analog domain, increasing dynamic range has severe energy cost compared to the digital domain [19]. To overcome these challenges, the MMADC takes the approaches described below.

A. Passive Analog Multiplication

The idea of performing multiplication within ADC has recently been explored [20], [21]. Different from previous approaches, the MMADC performs multiplication using passive circuits by exploiting attenuation in a negative feedback loop. The block diagram of a standard SAR ADC is shown in Fig. 10(a). Conversion is achieved through a clocked capacitive DAC (capDAC) within a discrete-time negative feedback loop. With the comparator providing high gain A, the overall transfer function, is unity, yielding a digital code equivalent to the sampled analog input

$$H_{SAR} = \frac{A}{1+A} \approx 1. \tag{6}$$

The MMADC extends the SAR architecture by adding a passive Feedback Divider (FB-Divider) block, which introduces a gain of β following the capDAC. The FB-Divider causes attenuation through a configurable capacitive voltage divider. Thus, the overall transfer function is

$$H_{MMADC} = \frac{A}{1+A\beta} \approx \frac{1}{\beta}, \tag{7}$$

yielding a digital code equivalent to the sampled analog input multiplied by $1/\beta$, a number larger than one. Thus multiplication is achieved passively through capacitors, which typically benefit from both better matching and better stability than active components. However, the maximum range of the sampled analog input is limited to the maximum range of the feedback loop, under the largest attenuation condition. To address

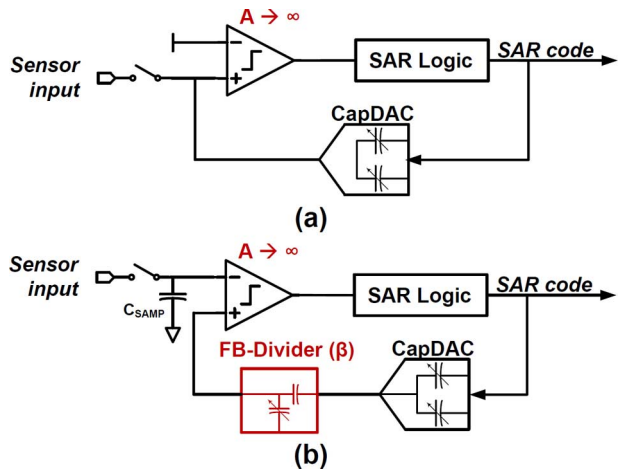


Fig. 10. The block diagrams showing (a) a standard SAR ADC, and (b) the MMADC, incorporating the feedback divider (FB-Divider) block to enable passive multiplication.

the reduced input analog range that occurs with large valued multipliers $1/\beta$, mixed-signal floating-point multiplication is employed.

B. Mixed-Signal Floating-Point Multiplication

As shown in (8), the MMADC breaks the effective multiplier M_{QUANT} (corresponding to quantized $h_{i,j}$) into three elements: (1) a sign bit M_S; (2) a significand M_A; and (3) an exponent M_D, with base 2.

$$M_{QUANT} = M_S \cdot M_A \cdot 2^{M_D} \tag{8}$$

The MMADC block diagram in Fig. 9 shows the hardware used to apply each of these. As seen, only M_A is applied via the FB-Divider. In the implementation presented, M_A is a 4-bit number designed to give values of β between 1 and $1/2$ in 16 steps: $\{16/16+0, 16/16+1, \ldots, 16/16+15\}$. Thus, analog multiplication is only performed by a significand from 1 to 2. Following SAR conversion and multiplication by the significand, the MMADC applies the sign bit M_S and multiplies by 2^{M_D} in the digital domain, simply through barrel shifting. In the implementation presented, M_D is a 5-bit number, enabling multiplication by a number between 2^{-16} to 2^{15}. In this way, the overall multiplier value M_{QUANT} can take on extremely large values (up to 64 k), but the analog dynamic range is affected by a factor less than 2.

Of course, such a floating-point multiplication impacts the quantization error, changing the SQNR of the overall multiplication operation. In fact, such a scheme substantially improves the overall SQNR. This is because the output quantization levels essentially shift based on the M_D value to a range around the multiplication result. On the other hand, with fixed-point multiplication, output quantization is fixed regardless of the multiplication result, causing the SQNR for low-valued results (which are obtained when the multiplier is small) to be degraded.

To illustrate this, Fig. 11 shows a simulation considering multiplication of an analog input X having infinite precision (to the

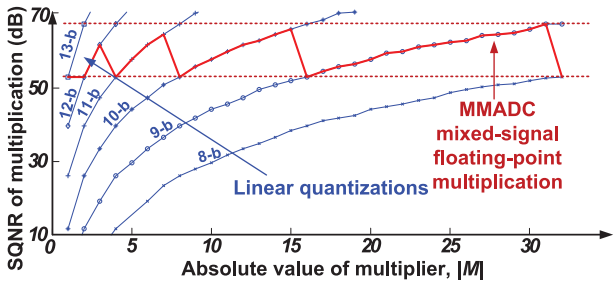


Fig. 11. The signal-to-quantization-noise ratio (SQNR) of linear fixed-point multiplication versus floating-point multiplication [4].

limit of the simulator) with a digital multiplier M. For this example, X is assumed to be uniformly distributed in the range of $[0, 1]$, while for illustration integer values of M up to 32 are considered. The blue lines show the SQNR for different fixed-point quantization levels of the output multiplication result, from 8 to 13 bits. As seen, the SQNR varies substantially with the multiplier value. In particular, small-valued multipliers lead to small-valued outputs; though, with fixed-point linear quantization, the output quantization error remains constant. Consequently, the SQNR is low for low-valued multipliers. In many applications, including the ones demonstrated in Section VI, low-valued multipliers occur with high frequency. This implies inefficient use of the hardware's dynamic range.

However, with the proposed floating-point multiplication, the SQNR essentially jumps between the linear fixed-point quantization curves as the M_D value changes. Namely, in the MMADC different values of M_D cause barrel shifting of the output result, while different values of M_A cause the SQNR to follow the linear fixed-point quantization curves. With barrel shifting in this way, M_D causes both the quantization error and the multiplier value to scale similarly. Since scaling of the multiplier value causes the output result to also scale, the overall SQNR remains more constant over a large range of multiplier values. Thus, more efficient use of the hardware's dynamic range is achieved.

C. Optimization for Reducing Analog Sensitivities

Even though the approach above allows us to limit the range of M_A and thus minimize the impact on the analog dynamic range, generally larger values of M_A are preferred to further reduce sensitivities to analog imprecisions. As an illustration, we can see from Fig. 9 that larger values of M_A (corresponding to smaller values of β), lead to larger signals at the input of the comparator. The floating-point multiplication used in the MMADC, enables a simple optimization knob by which this can be achieved.

In [22] an optimization was presented for minimizing the quantization error when mapping a set of multipliers to finite-precision floating-point representation. This is based on the insight shown in Fig. 12. With floating-point representation, a fixed-sized set of quantization levels are linearly spaced, and then the set is repeated by exponentially increasing the spacing. This exponential spacing means that the multipliers, regardless of their original values, can all be scaled by a single factor α to shift them closer to nearby quantization levels. The impact on

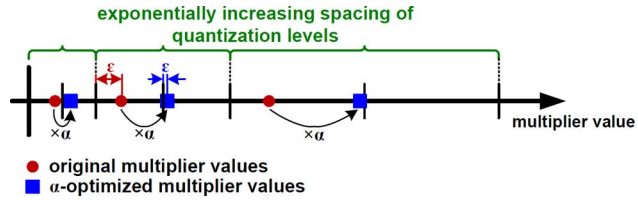


Fig. 12. When mapping a set of multipliers to floating-point representation, a single scaling factor α can be used to shift the values closer to the quantization levels, thereby reducing quantization error [22].

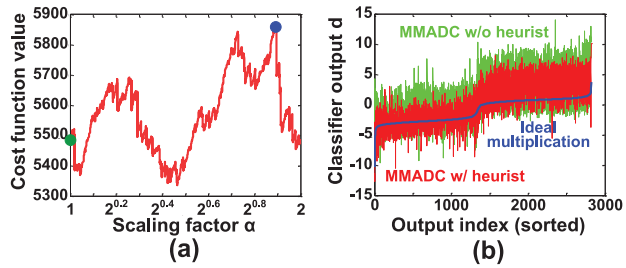


Fig. 13. Analysis of the proposed optimization, showing (a) the cost function value for $\alpha \in [1, 2]$, with both the optimal point (blue dot) and the default value without optimization (green dot), and (b) comparison of the classifier output, comparing ideal multiplication (blue), MMADC-measured multiplication with the optimization (red), and MMADC-measured multiplication without the optimization (green).

the final result is simply an overall scaling by α, which is typically easily tolerated.

The optimization of interest for the MMADC is to shift the multipliers such that most of them map to large-valued quantization levels among those linearly spaced within a given set. This is shown in Fig. 12 for the case of only two quantization levels within the linearly-spaced set, i.e., one-bit M_A; in the MMADC implementation presented, M_A is actually four bits. Since the rows of the $\mathbf{H}$ matrix each correspond to a linear classifier, the optimal α can be determined for each row. Further, this can be done off line following the training process, by employing the same data used for training. The objective function we define for performing this optimization is as follows:

$$\max_{\alpha \in R^+} \frac{1}{\alpha} \sum_{1}^{N} |\alpha M_{QUANT}| \cdot M_A(\alpha) \tag{9}$$

where

$$M_A(\alpha) = \frac{|\alpha M_{QUANT}|}{2^{\lfloor log_2 |\alpha M_{QUANT}| \rfloor}}. \tag{10}$$

As described in [22], such an objective function exhibits periodicity, implying that the globally-optimal solution can be found within the range of $\alpha \in [1, 2]$. For illustration, Fig. 13 a plots the objective function for a set of linear classifier coefficients obtained from training. We can see that the optimal α is approximately $2^{0.89}$ (1.85), which is larger than the default value of $\alpha = 1$ obtained if no such optimization is performed. Fig. 13(b) compares the corresponding classifier outputs d for the training data, considering ideal double-precision multiplication (blue curve), MMADC hardware measurement results with the

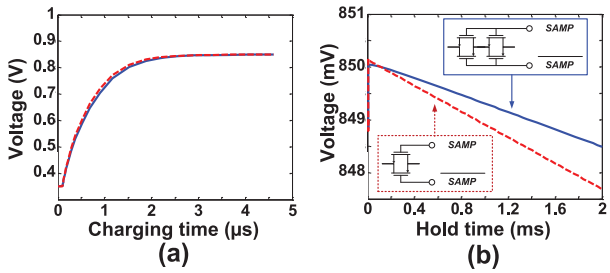


Fig. 14. Comparison of input sampling (a) charging and (b) holding time with (blue solid) and without (red dashed) stacked effect.

optimization (red curve), and MMADC hardware measurement results without the optimization (green curve) (for visualization, the classifier outputs are sorted based on increasing value of the ideal output). We can see that the optimization substantially reduces error, with the RMS value of error normalized to the RMS value of the ideal output being reduced from 2.0 down to 1.2.

IV. Circuit Design

This section describes the circuit implementation of the MMADC, focusing on the block diagram shown in Fig. 9. As mentioned, the MMADC expands on a tradition SAR ADC, designed to provide 8-bit output codes in the implementation presented. The major difference between the MMADC and a conventional SAR ADC are the use of an explicit sample-and-hold (S/H) circuit, as well as the feedback divider (FB-Divider) and barrel-shifter blocks for implementing multiplication. Combining the sign bit, 5 b barrel shifting (0 to 31), and 8 b SAR ADC output, the total bit width of the output code needed is 40 b. These blocks are discussed in detail next, followed by analysis of the error sources resulting from the hardware implementation.

A. Input Sampling

In a SAR ADC, the S/H block is typically combined with the capDAC. However, as discussed in Section Section III, to implement matrix multiplication, multiple conversions are needed of each analog sample. This necessitates resetting the capDAC, which will cause charge from the input sample to be lost. Thus, in the implementation presented, the sample's charge is stored on a separated capacitor. Further, the charge must be held for a longer period spanning the multiple conversions. To mitigate charge leakage during this period, the sampling switch uses two series connected CMOS switches, as shown in Fig. 14. Such a connection exploits the stacked effect, which reduces the leakage current more than the on current [23]. Across all corners and input-sample bias levels, the worst-case charge leakage is simulated to be less than 0.5 LSB over 50 conversions, thus allowing the **H** matrix to have 50 rows (corresponding to 50 weak-classifier iterations, in the formulation of Section II). Bootstrapping techniques to reduce the distortion of input sampling are not employed in the implementation presented due to the focus on relatively low-bandwidth signals in the system demonstrations (Section VI); however, standard bootstrapping approaches can be employed [24], [25].

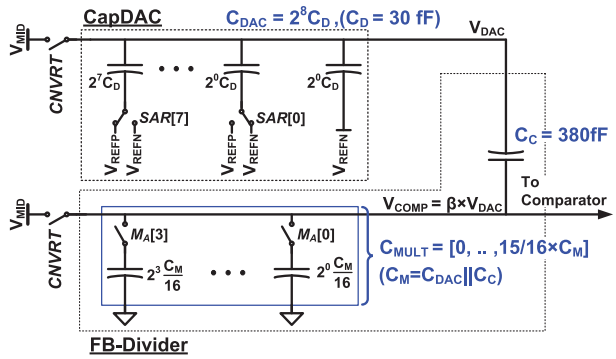


Fig. 15. The feedback divider block, consists of a coupling capacitor C_c and a variable capacitor C_{MULT}, the feedback gain β is as shown [4].

B. Feedback Divider (FB-Divider)

The FB-Divider block is used to implement attenuation via capacitive voltage division in the feedback path following the capDAC. The capDAC and FB-Divider circuits are shown in Fig. 15. The capDAC has a total capacitance of C_{DAC}; with binary-weighted unit capacitors $C_D = 30$ fF, C_{DAC} has a value of ~ 7.7 pF for the 8-bit DAC.

The capDAC output V_{DAC} drives the FB-Divider. The FB-Divider then performs voltage division between a coupling capacitor C_C and a switchable-valued capacitor C_{MULT}, yielding the output voltage C_{COMP} to the comparator. C_{MULT} is formed from four selectable binary-weighted capacitors, based on the 4-bit M_A value. The unit capacitor for binary weighting is designed to have a nominal value of $C_M/16$, where $C_M = C_{DAC}||C_C$. With C_c set to $\sim$ 380 fF, the unit capacitor is roughly 23 fF. Setting the capacitances in this way, the feedback gain β, applied to the ideal capDAC output, is given as follows:

$$\beta = \frac{C_{DAC}||C_C}{C_{DAC}||C_C + C_{MULT}} \tag{11}$$

$$= \frac{C_M}{C_M + C_{MULT}}. \tag{12}$$

Thus, with C_{MULT} taking values from 0 to $15/16 C_M$ (in 16 steps of $C_M/16$), β takes values between 1 and $16/31$, implementing maximum attenuation just less than $1/2$.

At the start of the conversion, the $CNVRT$ signal is asserted and the bottom plates in the capDAC are biased as shown in Fig. 15. This causes the charge of the capDAC and FB-Divider capacitors to be reset, setting V_{DAC} and V_{COMP} to a nominally mid-rail voltage V_{MID}. With the capDAC thus configured for MSB comparison, the analog input sample V_{IN} can be positive or negative with respect to V_{MID}. This, along with the M_S bit of the digital multiplier, allows the analog input and digital multiplier used by the MMADC to be signed. Thus, overall, the 8-bit output code $SAR[7:0]$ is derived to correspond to the following analog voltage:

$$(V_{IN} - V_{MID}) \cdot \frac{1}{\beta} \tag{13}$$

$$(V_{IN} - V_{MID}) \cdot \frac{C_M + C_{MULT}}{C_M} \tag{14}$$

$$(V_{IN} - V_{MID}) \cdot \frac{16 + M_A[3:0]}{16}. \tag{15}$$

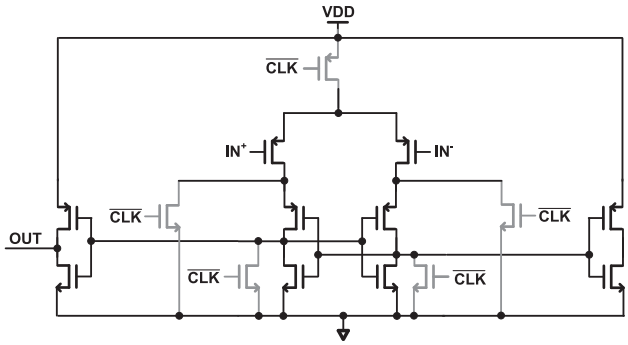


Fig. 16. Schematic of the comparator [26].

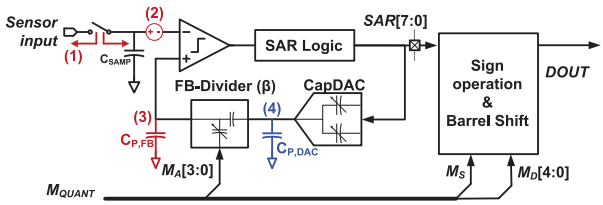


Fig. 17. Summary of error sources due to the modified SAR structure.

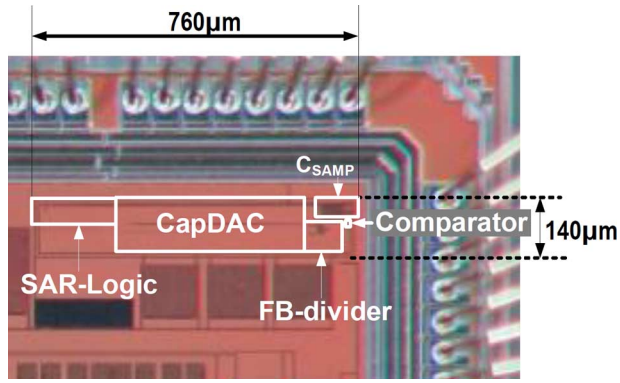


Fig. 18. Die photo of the MMADC prototype [4].

TABLE II
SUMMARY OF MEASUREMENT RESULTS [4]

Technology	130nm CMOS	Sample Rate	Up to 20k Hz
Supply Voltage	1.2V Analog 1.2V Digital	SNDR/ENOB	45.8dB / 7.31b (1kHz) 38.9dB / 6.17b (9kHz)
Resolution	8 bits	DNL/INL	0.30 / 0.32 LSB
Input Range	350mV - 850mV	FOM	209fJ / c.-s. (1kHz)

C. Comparator

The comparator is based on a dynamic latch structure, as shown in Fig. 16 [26]. At every positive edge of $\overline{CLK}$, the latch resets. At negative edges, the latch dynamically regenerates depending on the differential inputs.

D. Error Sources

Compared to a conventional SAR ADC, whose primary error sources at the 8-bit level are capacitor mismatch in the capDAC and distortion due to varying resistance of the input switch, the MMADC introduces additional error sources. These are summarized in Fig. 17 as $(1-3)$, along with an additional non-ideality (4), which does not lead to errors of importance.

The first two error sources (1, 2) are input dependent charge-injection error and input dependent comparator-offset, respectively. Both of these arise due to the fact that the S/H is separated from the capDAC. Done this way, the bias voltage of the sample switch varies with the input voltage leading to corresponding variations in the charge-injection error. Further, with the input sample serving as the reference voltage for comparison with the feedback path, the comparator biasing also changes with the input voltage. Typically, the comparator offset is dependent on its input biasing [27], thus leading to input-dependent conversion error. The third error source (3) arises due to parasitic capacitance on the top plate of the FB-Divider. Considering (12), this adds a fixed term to the denominator, causing error in the overall β value. Measurements in Section V show that the resulting error is modest, and can be further reduced through appropriate sizing of the comparator input devices and the use of FB-Divider capacitors with low top-plate parasitics.

The remaining non-ideality noted (4) arises due to parasitic capacitance on the top plate of the capDAC. Such a parasitic is also seen in SAR ADCs that employ a sub-DAC; such capacitance is easily handled by appropriately adjusting the value of the coupling capacitor C_C [27]. In the case of the MMADC, this can be seen in (12), where C_{DAC} only appears in terms with C_C (i.e., $C_{DAC}||C_C$).

Overall, measurement results in Section V show that the error sources have modest impact. Further, within an application these are readily overcome using the EACB approach previously described (Section II-C).

V. PROTOTYPE MEASUREMENTS

The MMADC prototype is fabricated in an IBM 130 nm CMOS technology [4]. Fig. 18 shows the die photo, and Table II provides a summary of the measurement results. The total MMADC area is 760 μm $\times$ 140 μm, with the FB-Divider (implemented) and digital barrel shifter (from post-layout area estimation) representing less than 5% area overhead compared to a standard SAR layout. The IC operates from a 1.2 V supply; however, the analog input range is from 0.35 V to 0.85 V, reduced by roughly a factor of 2, due to the analog-domain multiplication. The ADC power is seen to scale linearly with the sampling rate, giving roughly constant energy. The maximum sampling rate is 20 kHz. With an input signal of 1 kHz, a figure-of-merit (FoM) of 209 fJ per conversion-step is achieved. Beyond static non-linearity, the primary degradation of SNDR (45.8 dB with 1 kHz input, 38.9 dB with 9 kHz input) is due to distortion from the input switch; as mentioned, this can be addressed through bootstrapping techniques. We also point out that, based on recent designs reported, the SAR architecture enables the potential for signficant FoM reduction through various optimizations (e.g., amenability to scaled technologies, lower supply voltages, improved matching of unit capacitors). For the most part, these can readily be applied to the MMADC, giving the potential for further energy reductions.

For INL/DNL testing, different values of M_A result in different range of codes that can be reached (over the 0.35–0.85 V analog input range). For characterization over the entire range, Fig. 19 shows the INL/DNL with M_A set to 4'b1111. As seen,

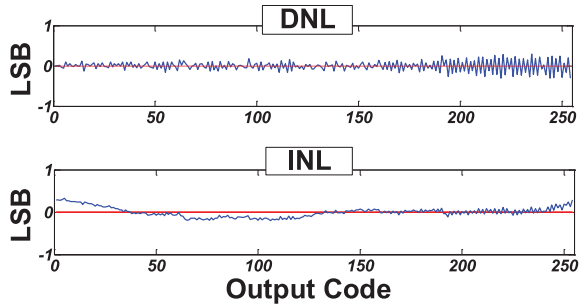


Fig. 19. INL and DNL of the MMADC with M_A set to 4'b1111 [4].

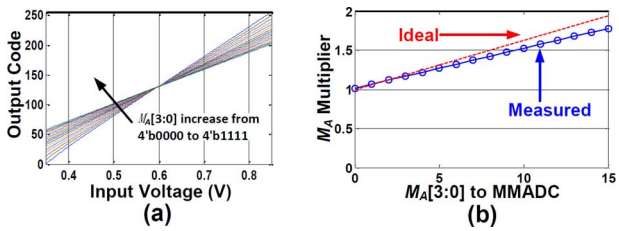


Fig. 20. Measured multiplication transfer function [4].

the worst case INL and DNL are 0.30 LSB and 0.32 LSB, respectively.

The measured multiplication transfer function is shown in Fig. 20. Fig. 20(a) shows the output code versus input voltage for different values of M_A. Fig. 20(b) uses these results to derive the overall multiplication transfer function verses M_A. As mentioned in Section IV-D, parasitic capacitance on the top plate of the FB-Divider causes some error. This error can be attributed to any value of M_A by selecting an overall scaling factor; in the plot shown, the error is attributed to largest M_A, amounting to 9% worst-case error for the capacitance values and MIM capacitors used. Note, application of M_D simply involves digital barrel shifting, and thus introduces no error beyond the quantization noise analyzed in Section III-B. Over all of the multiplications performed in the system demonstrations presented next, the RMS error of multiplication, normalized to the RMS of ideal multiplication is just 0.018. Thus, the MMADC provides robust computation combined with data conversion.

VI. System Demonstrations

Two applications have been demonstrated using the MMADC: (1) ECG-based cardiac arrhythmia detection; and (2) image-pixel-based gender detection. These applications, as well as their implementation and characterization details, are discussed in the following subsections.

A. ECG-Based Cardiac Arrhythmia Detection

Cardiac arrhythmias, in addition to causing considerable adverse symptoms, are also an indicator of a range of other important diseases. For this reason, arrhythmia detection is gaining prominence in patient-monitoring applications. The ECG presents strong biomarkers for detection, and a variety of methods for detection based on the ECG have been proposed [28], [29], [7].

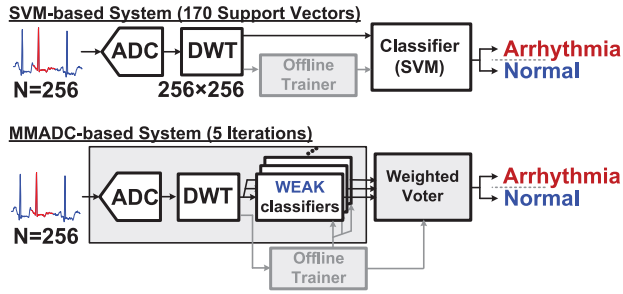


Fig. 21. ECG-based cardiac arrhythmia detection systems, showing conventional system (above) and proposed system (below). The systems correspond to the following parameter values: $N = 256$, $J = 256$, $S = 170$, and $K = 5$ *to* 9.

Specifically, Fig. 21 shows the block diagrams corresponding to the demonstrations presented, including a conventional system implementation (based on an ADC, feature extractor, and RBF SVM classifier) and the proposed system implementation (MMADC and adder for weighted voting). The systems presented employ features extracted from a discrete wavelet transform (DWT) taken over 256 samples of the ECG (which is itself sampled at 256 Hz) [7]. The DWT corresponds to a seven-stage level-four DWT [7]. This can be reformulated into a 256×256 matrix. For the conventional system, the RBF SVM is trained and found to require 170 support vectors. Similarly, for the proposed system, boosted linear classifiers are trained. As we show, boosting enables a knob for energy and precision scaling, wherein fewer iterations can be used to reduce energy. Thus, referring to the parameters from Section II, $N = 256$, $J = 256$, $S = 170$, and $K = 5$ *to* 9.

The conventional system is simulated in MATLAB to the full precision of the simulator to characterize its performance and in NanoSim to estimate its energy (following transistor-level circuit design and layout); the proposed system is implemented and tested experimentally. Expert-annotated ECG data of patients (identifying bundle branch block beats, atrial premature beats, nodal premature beats, premature ventricular contractions, atrial escape beats, nodal escape beats, supraventricular escape beats, ventricular escape beats and paced beats) are obtained from the MIT-BIH arrhythmia database [5], [6]. For the experimental demonstration, the signals are replayed using a 14-bit arbitrary waveform generator and provided to the MMADC.

1) System Performance: The metrics used to measure classification performance are true-positive rate and true-negative rate (TPR/TNR). TPR corresponds to the number of correctly classified testing-data instances from the positive class, divided by the total number of testing-data instances from the positive class (TNR is defined similarly). The false-positive/false-negative rates (FPR/FNR) can be derived as $\mathrm{FPR} = 1 - \mathrm{TNR}$ and $\mathrm{FNR} = 1 - \mathrm{TPR}$, respectively. For system training and testing, 10-fold cross validation is used. Fig. 22(a) and (b) compare the classification performance for the conventional system, the MMADC system with EACB (described in Section II-C) and with the multiplier optimization (described in Section III-C), the MMADC with EACB but without the optimization, and the MMADC without EACB and with the optimization. As seen, the MMADC system, together with EACB and the optimization,

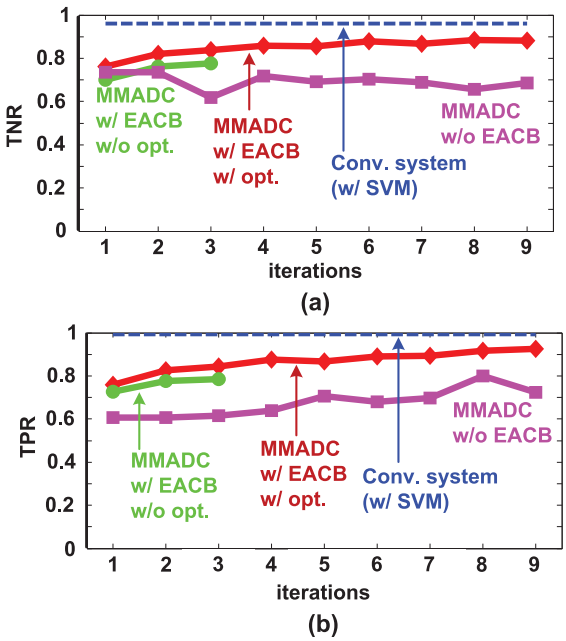


Fig. 22. ECG arrhythmia detection system performance [(a) true negative rate and (b) true positive rate] comparison for conventional system (blue), MMADC system without EACB (pink) and MMADC system with EACB without (green) and with (red) the heuristic.

is able to approach the performance of the ideal MATLAB-implemented conventional system within a small number of weak classifier iterations. We point out that various algorithms employing different features and classifiers have been proposed for arrhythmia detection, giving rise to various trade-offs in detection [30]. The performance of the systems considered here are representative of the performance levels that have been reported. Though some small performance difference remains between the MMADC sysetm and the conventional systems, such a system can be adequate for very-low-energy initial detection. The benefits of EACB, for improving accuracy, and the optimization, for improving boosting convergence are apparent.

2) System Energy: Though the performance levels of the conventional and MMADC systems are similar, the energy of the MMADC system, particularly when employing the system formulation presented in Section II, can be substantially lower. For energy comparison, the conventional system is represented as follows. The energy per ADC conversion ($E_C = 33.2$ pJ) is taken to be that measured from the MMADC, since the multiplication functionality added has negligible impact on energy. The feature-extraction energy, which corresponds to matrix multiplication, is estimated from post-layout simulation of a multiply-accumulate unit [13 b multiplication ($E_\times = 3.79$ pJ), 40b addition ($E_+ = 0.92$ pJ)] in the same 130 nm CMOS technology. The classifier energy, which corresponds to computation of the RBF SVM kernel (1), is estimated from post-layout simulation of the multiply-accumulate unit, a 40-bit adder, and a CORDIC unit ($E_e = 23.7$ pJ) [31], all implemented in the same 130 nm technology. For energy comparison, the MMADC system is estimated as follows. The MMADC energy, which corresponds to A-D conversion with matrix element multiplication, is measured from the prototype. Accumulation for matrix multiplication is estimated from post-layout simulation of an adder. Weighted voting, which corresponds to signed addition, is also estimated from post-layout simulation of an adder in the same 130 nm technology.

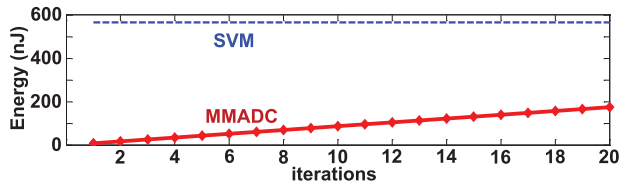


Fig. 23. Energy comparison of conventional and MMADC systems with increasing weak classifier iterations.

TABLE III
ENERGY AND PERFORMANCE COMPARISON FOR ECG-BASED ARRHYTHMIA-DETECTION SYSTEMS

Conventional System		MMADC System	
ADC	$E_C\ N = 8.50nJ$	ADC	$E_C\ N\ K = 42.5nJ$
F. E.	$J\ N\ E_\times + J\ (N-1)\ E_+ = 308nJ$	Acc.	$K\ (N-1)\ E_+ = 1.17nJ$
Cla.	$J\ S\ E_\times + 2\ J\ S\ E_+ + S\ E_e = 249nJ$	Vote	$(K-1)\ E_+ = 3.68pJ$
Total	$566nJ$	Total	$43.7nJ$
TPR/TNR	0.99/0.96	TPR/TNR	0.93/0.89

Fig. 23 compares the energy of the conventional system with MMADC system. While the energy for the MMADC system scales with weak classifier iterations, it remains much lower than the conventional system, even at 20 iterations. Using the system parameters with $K = 5$, the energies of the conventional and MMADC systems are shown in detail in Table III. The conventional system achieves an energy per classification of 566 nJ, while the MMADC system achieves an energy per classification of 43.7 nJ, representing 13× lower energy.

3) Quantization Error: As discussed in Section III-B, the MMADC uses a mixed floating-point (for M) and fixed-point (for X) multiplication. It was mentioned that conventional linear fixed-point multiplication leads to degraded SQNR for low-valued multipliers, leading to inefficient use of the hardware's dynamic range. For the arrhythmia-detection application, Fig. 24(a) normalizes the multiplier values (corresponding to entries of the $\mathbf{H}$ matrix) to $[-16, 16]$, and plots their histograms. Indeed, we find that low-valued multipliers are most prominent. Further, Fig. 24(b) plots a histogram of the quantization error of the multiplier values using both linear fixed-point representation (at 5-bit precision) and the floating-point representation of the MMADC (with 1 bit for M_S and 4 bits for M_A). As seen, the fixed-point representation has much more uniformly distributed error between $[0, 1]$, while the floating-point representation has error distributed much more densely near smaller values. This motivates the multiplication approach of the MMADC, particularly since similar distributions are observed more broadly. As an example, the multipliers involved in window-based FIR filtering matrices, which are commonly used in signal-processing systems, exhibit similar distribution [22].

B. Image-Pixel-Based Gender Detection

Gender detection from facial images is used widely in image-processing systems for applications ranging from biometrics

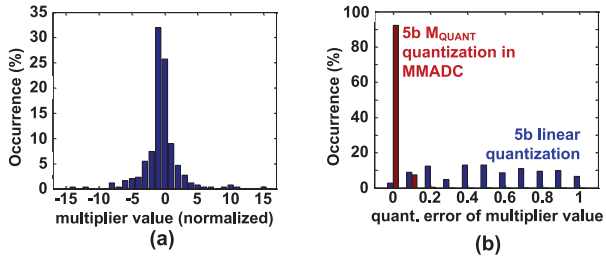


Fig. 24. Analysis of quantization errors in the ECG-based arrhythmia-detection system, showing (a) histogram of multiplier values normalized to 5 bits, and (b) corresponding histogram of quantization errors with 5-bit linear quantization and 5 bit M_S, M_A floating-point quantization, as applied in the MMADC.

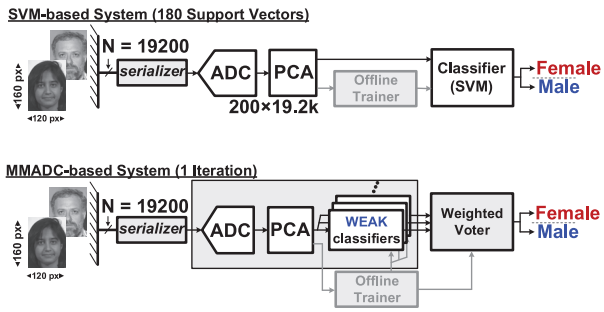


Fig. 25. Image-pixel-based gender-detection systems, showing conventional system (above) and proposed system (below). The systems correspond to the following parameter values: $N = 19{,}200$, $J = 200$, $S = 180$, and $K = 1$.

[32] to public security [33]. Fig. 25 shows block diagrams corresponding to the conventional-system and MMADC-system demonstrations presented. Each facial image is an aligned and sized to 160×120 pixels. Feature extraction corresponds to principal component analysis (PCA) applied to the 19,200-dimensional image-pixel vector. The first 200 principal components are referred to as projections onto eigen faces, and correspond to the feature vector used for classification [34]. Thus, feature extraction can be formulated into a 200×19200 matrix. For the conventional system, the RBF SVM is trained and found to require 180 support vectors. The proposed system, using boosted linear classifiers is found to require just one iteration. Thus, referring to the parameters from Section II, $N = 19{,}200$, $J = 200$, $S = 180$, and $K = 1$.

The conventional and proposed systems are simulated and measured as described previously. Application data is obtained from the FERET facial-image dataset [35]. Once again, for experimental demonstration of the proposed system, the image data is replayed through a 14-bit arbitrary waveform generator and provided one pixel at a time to the MMADC.

1) System Performance: Once again using the metrics of TPR/TNR, Fig. 26 shows the performance of the conventional system and the MMADC system (for training and testing 2-fold cross validation is used). As seen once again, the MMADC system achieves performance at the level of the ideal MATLAB-implemented conventional system. Further, the performance of both the systems considered is in the range of the state-of-the-art performance that has been reported for this application [36].

2) System Energy: The energies of the systems are estimated as described previously. Using the system parameters (N, J, S, K), the energies of the conventional and MMADC systems are

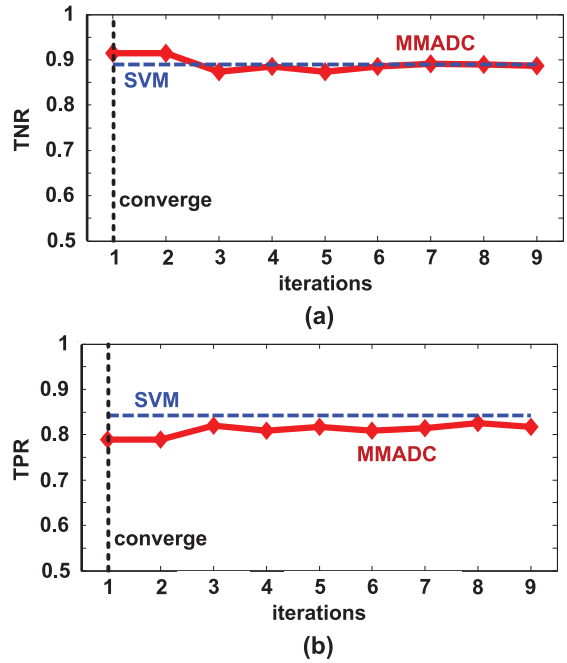


Fig. 26. image gender detection system performance comparison for conventional system and MMADC system [4].

TABLE IV
ENERGY AND PERFORMANCE COMPARISION FOR IMAGE-PIXEL-BASED GENDER-DETECTION SYSTEMS

Conventional System		MMADC System	
ADC	$E_C\ N = 637nJ$	ADC	$E_C\ N\ K = 637nJ$
F. E.	$J\ N\ E_\times + J\ (N-1)\ E_+ = 18.1\mu J$	Acc.	$K\ (N-1)\ E_+ = 17.7nJ$
Cla.	$J\ S\ E_\times + 2\ J\ S\ E_+ + S\ E_e = 207nJ$	Vote	$(K-1)\ E_+ = 0J$
Total	$18.9\mu J$	Total	$655nJ$
TPR/TNR	0.84/0.89	TPR/TNR	0.83/0.89

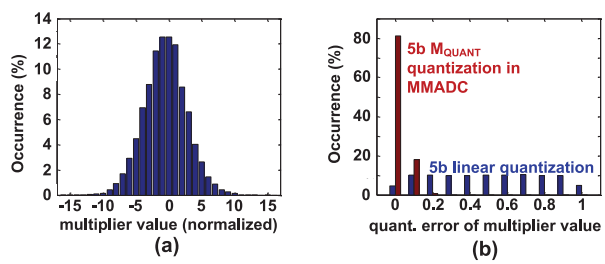


Fig. 27. Analysis of quantization errors in the image-pixel-based gender-detection system, showing (a) histogram of multiplier values normalized to 5 bits, and (b) corresponding histogram of quantization errors with 5-bit linear quantization and 5-bit M_S, M_A floating-point quantization, as applied in the MMADC.

shown in detail in Table IV. The conventional system achieves an energy per classification of 18.9 μJ, while the MMADC system achieves an energy per classification of 655 nJ, representing 29× lower energy.

3) Quantization Error: The quantization error incurred in this application is analyzed as described previously. As shown in Fig. 27(a), the histogram of multiplier values exhibits the same behavior, with low-valued multipliers occurring most frequently. As shown in Fig. 27(b), the quantization error of the multiplier values are once again distributed uniformly for the fixed-point representation and densely near smaller values for the floating-point representation used in the MMADC.

VII. Conclusions

Machine-learning algorithms for classification raise distinct benefits for analyzing physically complex embedded signals, such as those derived from medical sensors. For wearable and implantable systems, realizing such algorithm at minimal energy levels is of critical importance. This paper presents an algorithmic formulation that leverages classification via boosted linear classifiers, such that feature extraction and classification can be combined into a single matrix transformation. This eliminates explicit computations required to compute the feature vector and thus potentially leads to substantial energy savings. Further, this paper presents a matrix-multiplying ADC (MMADC) that enables this transformation to be largely performed within conversion process of a SAR ADC, by treating analog input samples as the elements of a source vector. When implementing systems in this way, a tradeoff is introduced between the dimensionality of the original feature vector and the number of linear classifiers required. Typically, the number of linear classifiers can be small leading to substantial energy savings. Further, recent work has shown that the energy of SAR ADCs can be substantially reduced, with designs today in the 6 to 9-bit range routinely achieving an energy per conversion step < 10 fJ. This makes the integration of multiplication functions in the ADC even more compelling, enabling increasingly complex computations to be performed at minimal additional energy.

References

[1] E. Jovanov, D. Raskovic, J. Price, A. Krishnamurthy, J. Chapman, and A. Moore, "Patient monitoring using personal area networks of wireless intelligent sensors," *Biomed. Sci. Instrum.*, vol. 37, pp. 373–378, 2001.

[2] T. Gao, D. Greenspan, M. Welsh, R. R. Juang, and A. Alm, "Vital signs monitoring and patient tracking over a wireless network," in *Proc. IEEE Int. Conf. Eng. Medicine and Biology Soc.*, 2006, pp. 102–105.

[3] V. Misra *et al.*, "Flexible technologies for self-powered wearable health and environmental sensing," *Proc. IEEE*, vol. 103, no. 4, pp. 665–681, 2015.

[4] J. Zhang, Z. Wang, and N. Verma, "A matrix-multiplying ADC implementing a machine-learning classifier directly with data conversion," in *Proc. IEEE Int. Solid-State Circuits Conf., Dig. Tech. Papers*, 2015, pp. 1–3.

[5] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Eng. Med. Biol. Mag.*, vol. 20, no. 3, pp. 45–50, 2001.

[6] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. 215–220, 2000.

[7] E.D. Übeyli, "ECG beats classification using multiclass support vector machines with error correcting output codes," *Digital Signal Process.*, vol. 17, no. 3, pp. 675–684, 2007.

[8] A. H. Shoeb and J. V. Guttag, "Application of machine learning to epileptic seizure detection," in *Proc. Int. Conf. Mach. Learning*, 2010, pp. 975–982.

[9] K. J. Jang, G. Balakrishnan, Z. Syed, and N. Verma, "Scalable customization of atrial fibrillation detection in cardiac monitoring devices: Increasing detection accuracy through personalized monitoring in large patient populations," in *Proc. IEEE Int. Conf. Eng. Medicine and Biology Soc.*, 2011, pp. 2184–2187.

[10] G. D. Magoulas and A. Prentza, "Machine learning in medical applications," in *Mach. Learning and its Applicant*. New York, NY, USA: Springer, 2001, pp. 300–307.

[11] A. Csavoy, G. Molnar, and T. Denison, "Creating support circuits for the nervous system: considerations for 'brain-machine' interfacing," in *Proc. IEEE Symp. VLSI Circuits*, 2009, pp. 4–7.

[12] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[13] R. Genov and G. Cauwenberghs, "Kerneltron: support vector 'machine' in silicon," *IEEE Trans. Neural Netw.*, vol. 14, no. 5, pp. 1426–1434, 2003.

[14] R. Genov, S. Chakrabartty, and G. Cauwenberghs, "Silicon support vector machine with on-line learning," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 17, no. 3, pp. 385–404, 2003.

[15] R. E. Schapire and Y. Freund, *Boosting: Foundations and Algorithms*. Cambridge, MA, USA: MIT Press, 2012.

[16] Z. Wang, R. Schapire, and N. Verma, "Error-adaptive classifier boosting (EACB): exploiting data-driven training for highly fault-tolerant hardware," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing*, 2014, pp. 3884–3888.

[17] Z. Wang, R. E. Schapire, and N. Verma, "Error adaptive classifier boosting (EACB): leveraging data-driven training towards hardware resilience for signal inference," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 4, pp. 1136–1145, 2015.

[18] W. Rieutort-Louis, T. Moy, Z. Wang, S. Wagner, J. Sturm, and N. Verma, "A large-area image sensing and detection system based on embedded thin-film classifiers," in *Proc. IEEE Int. Solid-State Circuits Conf., Dig. Tech. Papers*, 2015, pp. 1–3.

[19] R. Sharpeshkar, *Ultra low Power Bioelectronics: Fundamentals, Biomedical Applications, and Bio-Inspired System*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[20] H. Jafari, L. Soleymani, and R. Genov, "16-channel CMOS impedance spectroscopy DNA analyzer with dual-slope multiplying ADCs," *IEEE Trans. Biomed. Circuits Syst.*, vol. 6, no. 5, pp. 468–478, Oct. 2012.

[21] J. Zhang, L. Huang, Z. Wang, and N. Verma, "A seizure-detection IC employing machine learning to overcome data-conversion and analog-processing non-idealities," in *Proc. IEEE Custom Integrated Circuits Conf.*, 2015.

[22] Z. Wang, J. Zhang, and N. Verma, "Reducing quantization error in low-energy FIR filter accelerators," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing*, 2015.

[23] S. Narendra, V. De, D. Antoniadis, A. Chandrakasan, and S. Borkar, "Scaling of stack effect and its application for leakage reduction," in *Proc. Int. Symp. Low Power Electronics and Design*, 2001, pp. 195–200.

[24] N. Verma and A. P. Chandrakasan, "A 25 μW 100 kS/s 12 b ADC for wireless micro-sensor applications," in *Proc. IEEE Int. Solid-State Circuits Conf., Dig. Tech. Papers*, 2006, pp. 222–223.

[25] M. Dessouky and A. Kaiser, "Very low-voltage digital-audio $\Delta\Sigma$ modulator with 88-dB dynamic range using local switch bootstrapping," *IEEE J. Solid-State Circuits*, vol. 36, no. 3, pp. 349–355, 2001.

[26] T. Kobayashi, K. Nogami, T. Shirotori, Y. Fujimoto, and O. Watanabe, "A current-mode latch sense amplifier and a static power saving input buffer for low-power architecture," in *Proc. IEEE Symp. VLSI Circuits*, 1992, pp. 28–29.

[27] N. Verma, "An ultra low power ADC for wireless micro-sensor applications," M.S. thesis, Mass. Inst. Technol., Cambridge, MA, USA, 2005.

[28] A. Tayal, M. Tian, K. Kelly, S. Jones, D. Wright, D. Singh, J. Jarouse, J. Brillman, S. Murali, and R. Gupta, "Atrial fibrillation detected by mobile cardiac outpatient telemetry in cryptogenic TIA or stroke," *Neurology*, vol. 71, no. 21, pp. 1696–1701, 2008.

[29] J. J. Oresko, Z. Jin, J. Cheng, S. Huang, Y. Sun, H. Duschl, and A. C. Cheng, "A wearable smartphone-based platform for real-time cardiovascular disease detection via electrocardiogram processing," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 3, pp. 734–740, 2010.

[30] J. Wiens and J. V. Guttag, "Active learning applied to patient-adaptive heartbeat classification," in *Proc. Advances Neural Inf. Processing Syst.*, 2010, pp. 2442–2450.

[31] K. H. Lee and N. Verma, "A low-power microprocessor for data-driven analysis of analytically-intractable physiological signals in advanced medical sensors," in *Proc. IEEE Symp. VLSI Circuits*, 2013, pp. 250–251.

[32] W. Gao and H. Ai, "Face gender classification on consumer images in a multiethnic environment," in *Advances in Biometrics*. New York, NY, USA: Springer, 2009, pp. 169–178.

[33] K. W. Bowyer, "Face recognition technology: security versus privacy," *IEEE Technol. Soc. Mag.*, vol. 23, no. 1, pp. 9–19, 2004.

[34] S. Buchala, N. Davey, T. M. Gale, and R. J. Frank, "Principal component analysis of gender, ethnicity, age, and identity of face images," *IEEE Proc. ICMI*, 2005.

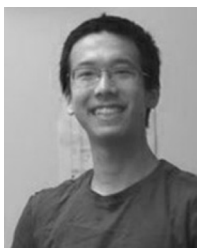
[35] P. J. Phillips *et al.*, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Trans. Pattern Anal. ach. Intell.*, vol. 22, no. 10, pp. 1090–1104, 2000.

[36] E. Mäkinen and R. Raisamo, "Evaluation of gender classification methods with automatically detected and aligned faces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 3, pp. 541–547, 2008.

Zhuo Wang (S'13) received the B.S. degree in microelectronics and the M.S. degree from Peking University, Beijing, China, in 2011 and 2013, respectively.

Currently, he is working toward the Ph.D. degree in the Department of Electrical Engineering, Princeton University, Princeton, NJ, USA. His research focuses on leveraging statistical approaches such as machine learning, for achieving hardware relaxation in an algorithmic and architectural level in resource-constrained platforms such as embedded sensing systems.

Mr. Wang was the recipient of 2011 Peking University Best Undergraduate Dissertation Award, 2014 ICASSP conference Best Paper Award nomination, and 2015 Princeton University Honorific Fellowship.



Jintao Zhang (S'12) received the B.S. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 2012.

Currently, he is working toward the Ph.D. degree at Princeton University, Princeton, NJ, USA. His research is focused on energy efficient signal analysis, mainly classification and on-chip. His primary research interests are VLSI system design, exploring mixed signal circuit design by using digital controls to enable various analog functionalities of the CMOS and thus enhancing the system performance with both benefit from analog circuits and digital circuits.

Naveen Verma (S'04–M'09) received the B.A.Sc. degree in electrical and computer engineering from the University of British Columbia, Vancouver, BC, Canada, in 2003, and the M.S. and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2005 and 2009, respectively.

Currently, he is an Associate Professor of electrical engineering at Princeton University, Princeton, NJ, USA, where he has been since 2009. His research focuses on advanced sensing systems, including low-voltage digital logic and SRAMs, low-noise analog instrumentation and data-conversion, large-area sensing systems based on flexible electronics, and low-energy algorithms for embedded inference, especially for medical applications.

Dr. Verma was a recipient or corecipient of the 2006 DAC/ISSCC Student Design Contest Award, 2008 ISSCC Jack Kilby Paper Award, 2012 Alfred Rheinstein Junior Faculty Award, 2013 NSF CAREER Award, 2013 Intel Early Career Award, 2013 Walter C. Johnson Prize for Teaching Excellence, 2013 VLSI Symposium Best Student Paper Award, and 2014 AFOSR Young Investigator Award.