

Embedded Algorithmic Noise-Tolerance for Signal Processing and Machine Learning Systems via Data Path Decomposition

Sai Zhang *Student Member, IEEE* and Naresh R. Shanbhag, *Fellow, IEEE*

Abstract—Low overhead error-resiliency techniques such as algorithmic noise-tolerance (ANT) have been shown to be particularly effective for signal processing and machine learning kernels. However, the overhead of conventional ANT can be as large as 30% due to the use of explicit estimator block. To overcome this overhead, embedded ANT (E-ANT) is proposed [1] where the estimator is embedded in the main computation block via data path decomposition (DPD). E-ANT reduces the logic overhead to be below 8% as compared with the 20% – 30% associated with conventional reduced precision replica (RPR) ANT system while maintaining the same error compensation functionality. DPD was first proposed in our original paper [1] where its benefits were studied in the case of a simple multiply-accumulator (MAC) kernel. This paper builds upon [1] by: 1) providing conditions for the existence of DPD, 2) demonstrating DPD for a variety of commonly employed kernels in signal processing and machine learning applications, and 3) evaluating the robustness improvement and energy savings of DPD at the system level for an SVM-based EEG seizure classification application. Simulation results in a commercial 45 nm CMOS process show that E-ANT can compensate for error rates up to 0.38 for errors in FE only, and 0.17 for errors in FE and CE, while maintain a true positive rate $p_{tp} > 0.9$ and a false positive rate $p_{fp} \leq 0.01$. This represents a greater than 3-orders-of-magnitude improvement in error tolerance over the conventional system. This error tolerance is employed to reduce energy via the use of voltage overscaling (VOS). E-ANT is able to achieve 51% energy savings when errors are in FE only, and up to 43% savings when errors are in both FE and CE.

I. INTRODUCTION

Error resiliency techniques have been proposed [2] to enhance energy efficiency by reducing design margins and compensating for the resultant errors. Large design margins arise from the need to provide robustness in the presence of process, voltage, and temperature variations [3], and represent an energy overhead as high as $3\times$ -to- $4\times$ [4]. The key to the use of error resiliency for energy reduction is that such techniques need to be low overhead and yet effective in compensating for high error rates. Classical fault-tolerance techniques such as N-modular redundancy (NMR) rely on replication of the main computation block and as a result are ineffective for

the purposes of energy reduction. Hence, low overhead error resiliency techniques such as RAZOR [5], [6], error-detection sequential (EDS) [7], confidence driven computing (CDC) [8] and statistical error compensation (SEC) techniques [2] have been proposed to enhance energy efficiency.

Error resiliency techniques have two sources of overhead: error detection and error correction. Techniques such as RAZOR [5], [6], EDS [7], and CDC [8] achieve lower error detection overhead compared with NMR. RAZOR employs a shadow latch for error detection with an error detection overhead of 1.2% (with 28.5% overhead in RAZOR flip flop). EDS [7] employs a dynamic transition detector with a time-borrowing data path latch (TDTB) or a double sampling static design with time-borrowing data path latch (DSTB) to achieve error detection and the error detection overhead is similar to RAZOR. CDC [8] employs fine-grained temporal redundancy and confidence checking with an error detection overhead of 8%–46%. However, these techniques employ a roll back based error correction scheme, thus incur error correction overhead of greater than $2\times$, and can compensate for error rates < 0.01 .

Unlike the roll back based techniques, SEC is a class of system level error compensation techniques that utilizes signal and error statistics and hence is particularly well-suited for signal processing and machine learning systems. These techniques include algorithmic noise tolerance (ANT), soft NMR, stochastic sensor network on a chip (SSNOC) [2], and have been shown to compensate for error rates ranging from 0.21 to 0.89, with an combined error detection and correction overhead ranging from 5% to 30% resulting in energy savings ranging from 35% to 72%.

ANT [2], is a specific SEC technique that has been shown to be effective in compensating for high error rates in signal processing and machine learning kernels. For example, the reduced precision replica (RPR) ANT technique and prediction based ANT was employed to compensate for error rates of $0.27 \sim 0.58$ in an ECG processor [9], [10] while delivering the required application-level performance. The overhead in ANT ranges from 5% to 30% [9] due to the use of explicit estimator blocks in error compensation. This overhead, though small compared to other techniques, limits the achievable system level energy efficiency between 28% \sim 41%.

In this paper, we propose embedded statistical error compensation where the main block is reformulated at the architectural level to embed an estimator for low overhead error detection and correction. In particular, embedded ANT (E-ANT) is proposed by embedding a RPR estimator into the

Copyright (c) 2015 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

Sai Zhang is with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, USA, e-mail: s Zhang12@illinois.edu.

Naresh R. Shanbhag is with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, USA, e-mail: shanbhag@illinois.edu.

main block via a technique called *data path decomposition* (DPD). Such decomposition is achieved by decomposing the main block into the most significant bit (MSB) and the least significant bit (LSB) components and employing the MSB component as the estimator output. As a result, DPD based E-ANT achieves the same error compensation functionality with lower complexity and improved energy savings as the conventional ANT technique. DPD was first proposed in our original paper [1] where its benefits were studied in the case of a simple multiply-accumulator (MAC) kernel. This paper builds upon [1] by: 1) providing conditions for the existence of DPD, 2) demonstrating DPD for a variety of commonly employed kernels in signal processing and machine learning applications (filter, butterfly unit, exponential kernel and others), and 3) evaluating the robustness improvement and energy savings of DPD at the system level for an SVM-based EEG seizure classification application.

E-ANT is applied to the design of an EEG seizure classification system [11] consisting of a frequency selective filter bank as the feature extractor (FE) and a support vector machine (SVM) as the classification engine (CE). Simulation results in a commercial 45 nm CMOS process show that E-ANT can compensate for error rates of up to 0.38 (errors in FE only), up to 0.17 (errors in FE and CE), and maintain the true positive rate $p_{tp} > 0.9$ and false positive rate $p_{fp} \leq 0.01$. This represents a greater than 3-orders-of-magnitude improvement in error tolerance over the conventional architecture. This error tolerance is employed to reduce energy via the use of voltage overscaling (VOS). E-ANT is able to achieve 51% energy savings when errors are in FE only, and up to 43% savings when errors are in both FE and CE.

The rest of the paper is organized as follows. Section II describes the background of the ANT technique and the SVM EEG classification system architecture. Section III presents E-ANT and derives E-ANT architectures for a wide class of common signal processing and machine learning kernels. Section IV presents the design optimization and application of E-ANT to the SVM EEG classification system. Conclusions are presented in Section V.

II. BACKGROUND

A. Conventional ANT

Conventional ANT incorporates a main block (**M**) and an estimator (**E**) as shown Fig. 1(a). The **M**-block implements the algorithm of interest and is conventionally error-free. In ANT, the **M**-block is permitted to make errors, which are then compensated for by the rest of the blocks in Fig. 1(a) including the **E**-block. In RPR ANT, the **E**-block is obtained by reducing the precision of the **M**-block. The **M**-block is subject to large magnitude errors η (e.g., timing errors due to critical path violations which typically occur in the MSBs) while the **E**-block is subject to small magnitude errors e (see Fig. 1(b), e.g., due to quantization noise in the LSBs), i.e.:

$$y_a = y_o + \eta \quad (1)$$

$$y_e = y_o + e \quad (2)$$

where y_o , y_a , and y_e are the error-free, the **M**, and **E**-block outputs, respectively. ANT exploits the difference in the error statistics of η and e to detect and compensate for errors to obtain the final corrected output \hat{y} as follows:

$$\hat{y} = \begin{cases} y_a & \text{if } |y_a - y_e| \leq T_h \\ y_e & \text{otherwise} \end{cases} \quad (3)$$

where T_h is an application dependent threshold parameter chosen to maximize the performance of ANT. In this paper, T_h is chosen to equal $\max(|y_o - y_e|)$ as this ensures that the **M**-block output y_a will always be selected [9] when the output is error free. The error rate p_η is defined as:

$$p_\eta = 1 - P_\eta(0) = Pr\{\eta \neq 0\} \quad (4)$$

where $P_\eta(\cdot)$ is the error probability mass function (PMF) of η . The errors η are most conveniently obtained by applying voltage overscaling (VOS) where the supply voltage V_{dd} is scaled as follows:

$$V_{dd} = K_{vos} V_{dd-crit} \quad (5)$$

where K_{vos} is the voltage overscaling factor, and $V_{dd-crit}$ is the minimum voltage needed for error free operation in the **M**-block. Note that for the ANT system to work properly, the **E**-block is not permitted to make large magnitude errors such as those arising from timing violations. This helps maintain the difference in the error statistics at the output of the **M** and **E**-block as shown in Fig. 1(b).

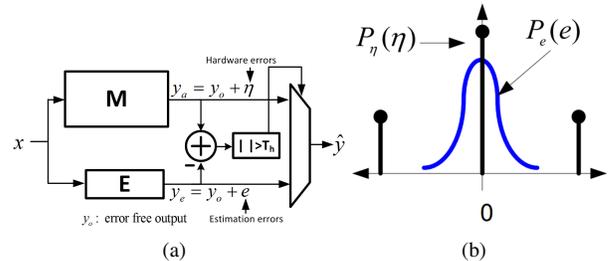


Figure 1. Algorithmic noise-tolerance (ANT): a) conventional architecture, and b) the error statistics in the main (**M**) and estimator (**E**) blocks.

The performance improvement achieved by ANT can be evaluated by employing a system level metric such as the signal-to-noise ratio (SNR). Assume that the error-free output in Fig. 1(a) is expressed as:

$$y_o = s + n_s \quad (6)$$

where s and n_s represent the signal and noise components in the error-free output y_o , respectively. At the application level, one is interested in the ratio of the signal power σ_s^2 to the noise powers at the outputs of the **M**, **E**-block and ANT system. Thus, the following application level SNRs can be defined:

$$SNR_{M,a} = 10\log_{10}\left(\frac{\sigma_s^2}{\sigma_{n_s}^2 + \sigma_\eta^2}\right) \quad (7)$$

$$SNR_{E,a} = 10\log_{10}\left(\frac{\sigma_s^2}{\sigma_{n_s}^2 + \sigma_e^2}\right) \quad (8)$$

$$SNR_{ANT,a} = 10\log_{10}\left(\frac{\sigma_s^2}{\sigma_{n_s}^2 + \sigma_{n_r}^2}\right) \quad (9)$$

where σ_s^2 , $\sigma_{n_s}^2$, σ_η^2 , σ_e^2 , $\sigma_{y_o}^2$ and $\sigma_{n_r}^2$ are the variances of the signal s , noise n_s , **M**-block hardware error η , **E**-block estimation error e , error-free output y_o , and residual error $n_r = y_o - \hat{y}$, respectively. It is also of interest to evaluate how 'noisy' the circuit fabric is with respect to an error-free (conventional) architecture. By definition, the output of such an architecture is y_o . Thus, we define the circuit level SNRs as follows:

$$SNR_{M,c} = 10\log_{10}\left(\frac{\sigma_{y_o}^2}{\sigma_\eta^2}\right) \quad (10)$$

$$SNR_{E,c} = 10\log_{10}\left(\frac{\sigma_{y_o}^2}{\sigma_e^2}\right) \quad (11)$$

$$SNR_{ANT,c} = 10\log_{10}\left(\frac{\sigma_{y_o}^2}{\sigma_{n_r}^2}\right) \quad (12)$$

If error detection is ideal, then $n_r \in \{0, e\}$, and its probability mass function (PMF) $P_{\eta_r}(n_r)$ is given by:

$$P_{\eta_r}(n_r) = \begin{cases} 1 - p_\eta & \text{if } n_r = 0 \\ p_\eta & \text{if } n_r = e \end{cases}$$

and

$$\sigma_{n_r}^2 = p_\eta \sigma_e^2$$

where p_η is the error rate of the **M**-block defined in (4). Therefore, (9) and (12) can be expressed as:

$$SNR_{ANT,a} = 10\log_{10}\left(\frac{\sigma_s^2}{\sigma_n^2 + p_\eta \sigma_e^2}\right) \quad (13)$$

$$SNR_{ANT,c} = 10\log_{10}\left(\frac{\sigma_{y_o}^2}{p_\eta \sigma_e^2}\right) \quad (14)$$

Since e is the small magnitude LSB error and η is the large magnitude MSB error, $p_\eta \sigma_e^2 \ll \sigma_e^2 \ll \sigma_\eta^2$. This further implies that $SNR_{ANT,a} \gg SNR_{E,a} \gg SNR_{M,a}$, and $SNR_{ANT,c} \gg SNR_{E,c} \gg SNR_{M,c}$. Thus, the output SNR of the ANT system is significantly greater than the SNR at the output of either the **M** or **E**-block. This phenomenon occurs in spite of the fact that the ANT system output $\hat{y} \in \{y_a, y_e\}$ (see Fig. 1(a)), i.e., \hat{y} equals the output of either the **M** or **E**-block. The reason for this unique feature of ANT is because it exploits the difference in the error statistics (see Fig. 1(b)) at the output of the **M** and **E**-block.

B. EEG classification system using SVM

Portable health monitoring is an important class of applications that can benefit from the design of energy efficient machine learning kernels. It has been shown [12] that epileptic seizures can be efficiently detected by analyzing the EEG signal using an SVM kernel. The EEG seizure classification system [12] shown in Fig. 2(a) consists of a frequency selective filter bank to extract signal energy in the 0 – 20 Hz range and a SVM classifier. The filter bank has passband of 3 Hz with a transition band of 1.5 Hz. Eight channels are employed to cover the entire frequency range [12].

SVM [13] is a popular supervised learning method for classification and regression. An SVM operates by first training the model (the training phase) followed by classification (the classification phase). During the training phase, feature vectors with labels are used to train the model. During the classification phase, the SVM produces a predictive label when provided with a new feature vector. The SVM training can be formulated as the following optimization problem to determine the maximum margin classifier [13] (see Fig. 2(b)):

$$\begin{aligned} \min \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_i \xi_i \\ \text{s.t.} \\ y_i (\mathbf{w}^T \mathbf{x}_i - b) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{aligned} \quad (15)$$

where C is the cost factor, ξ_i is the soft margin, \mathbf{x}_i is the feature vector, y_i is the label corresponding to the feature vector \mathbf{x}_i , \mathbf{w} is the weight vector, and b is the bias. The trained model is represented by:

$$y = \mathbf{w}_o^T \mathbf{x} - b \quad (16)$$

where \mathbf{w}_o are the optimized weights. It can be shown that the optimum weights are represented as a linear combination of the feature vectors that lie on the margins (see Fig. 2(b)), i.e., support vectors:

$$\mathbf{w}_o = \sum_{i=1}^{N_s} \alpha_i y_i \mathbf{x}_{s,i} \quad (17)$$

where N_s and $\mathbf{x}_{s,i}$ are the number of support vectors and i th support vector, respectively. The linear model can thus be represented as:

$$y = \sum_{i=1}^{N_s} \alpha_i y_i \mathbf{x}_{s,i}^T \mathbf{x} - b \quad (18)$$

The linear SVM in (18) can be easily extended into non-linear SVM by employing the kernel trick [13], resulting in:

$$y = \sum_{i=1}^{N_s} \alpha_i y_i K(\mathbf{x}_{s,i}, \mathbf{x}) - b \quad (19)$$

where $K(\mathbf{x}_{s,i}, \mathbf{x})$ is a kernel function. Popular kernel functions include polynomial, radial basis function (RBF), and others [14].

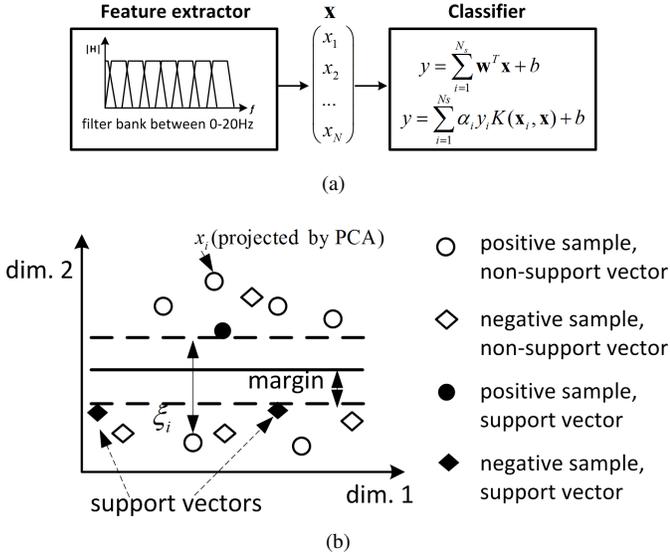


Figure 2. EEG seizure classifier with SVM: a) system architecture, and b) principle of SVM.

III. PROPOSED E-ANT TECHNIQUE

The key idea in E-ANT is to reuse part of the **M**-block to generate an estimate of its error-free output y_o . This is in contrast to conventional ANT, where an explicit **E**-block is required. We will show that such embedding of the **E**-block can be performed at the architectural level, where DPD is proposed to transform an existing architecture into an error resilient one.

A. DPD based E-ANT

In RPR ANT, the **M** and **E**-block process the same data but with different precisions. This redundancy can be exploited to embed the **E**-block into the **M**-block via DPD. In particular, DPD decomposes the **M**-block into MSB and LSB components, and employs the output of the MSB component as an estimate of the error-free **M**-block output y_o . By ensuring that the critical path of the MSB block is always shorter than that of the **M**-block, the requirements on the error statistics (see Fig. 1(b)) on the **M** and **E**-block are satisfied.

Let $y_a = f(x)$ denote the **M**-block functionality, where x and y_a are the input and output of the **M**-block, respectively. A B_x -bit input $x = x_0x_1\dots x_{B_x-1}$ can be written in 2's complement form [15], as follows:

$$\begin{aligned} x &= -x_0 + \sum_{i=1}^{B_x-1} x_i 2^{-i} \\ &= x_M + x_L 2^{-(B_{msb}-1)} \end{aligned} \quad (20)$$

where x_M is the value of B_{msb} MSB bits, and x_L is the value of $B_x - B_{msb}$ LSB bits, as shown below:

$$x_M = -x_0 + \sum_{i=1}^{B_{msb}-1} x_i 2^{-i} \quad (21)$$

$$x_L = \sum_{i=B_{msb}}^{B_x-1} x_i 2^{-(i-B_{msb}+1)} \quad (22)$$

Therefore, the **M**-block output is expressed as:

$$y_a = f(x) = f(x_M + x_L 2^{-(B_{msb}-1)})$$

In E-ANT, we decompose $f(x)$ as follows:

$$\begin{aligned} y_a &= f(x) \\ &= f(x_M + x_L 2^{-(B_{msb}-1)}) \\ &= g(f_M(x_M), f_L(x_M, x_L)) \end{aligned} \quad (23)$$

where $f_M(x_M)$ and $f_L(x_M, x_L)$ are functions that are combined by the operator $g(\cdot)$ to generate the final output y_a . Since this decomposition utilizes the finite precision nature of arithmetic units, it is referred to as DPD. We show that DPD exists if $f(x)$ is n -times differentiable or can be piecewise approximated. An E-ANT system can be obtained via DPD by ensuring that: 1) the critical path of $f_M(x_M)$ is shorter than that of $g(f_M(x_M), f_L(x_M, x_L))$, and 2) $f_M(x_M)$ generates an estimate y_e of the error-free output y_o . The operation of DPD based E-ANT is described as follows:

$$\begin{aligned} y_a &= g(f_M(x_M), f_L(x_M, x_L)) \\ y_e &= f_M(x_M) \\ \hat{y} &= \begin{cases} y_a & \text{if } |y_a - y_e| \leq T_h \\ y_e & \text{otherwise} \end{cases} \end{aligned}$$

where T_h is the error detection threshold as in (3). Next, we describe several methods to achieve DPD.

B. DPD via Taylor expansion

Taylor expansion can be employed to achieve DPD. If $f(x)$ is n -times differentiable in the input range $x \in [x_l, x_u]$. In the appendix A, we show that DPD for $f(x)$ using Taylor expansion is given by:

$$f(x) \approx f_M(x_M) + f_L(x_M, x_L) \quad (24)$$

where

$$\begin{aligned} f_M(x_M) &= f(x_0) \\ &\quad + \sum_{k=1}^n \sum_{i=0}^k \left[\frac{f^{(k)}(x_0)}{k!} \binom{k}{i} (-x_0)^{k-i} \right] x_{i,M} \\ f_L(x_M, x_L) &= \sum_{k=1}^n \sum_{i=0}^k \left[\frac{f^{(k)}(x_0)}{k!} \binom{k}{i} (-x_0)^{k-i} \right] x_{i,L} \\ x_{i,M} &= x_M^i \\ x_{i,L} &= \sum_{j=0}^{i-1} \binom{i}{j} x_M^j (x_L 2^{-(B_{msb}-1)})^{i-j} \end{aligned}$$

where x_M and x_L are defined in (21) and (22), respectively.

As a special case, when a first order Taylor expansion is employed at $x_0 = \frac{1}{2}(x_l + x_u)$, i.e., at center of the input dynamic range, (24) simplifies into:

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) \quad (25)$$

where $f'(x_0)$ is the first order derivative of $f(x)$ at x_0 . Substituting (20) into (25), we obtain the DPD of $f(x)$ as follows:

$$\begin{aligned} f(x) &\approx f(x_0) + f'(x_0)(x_M + x_L 2^{-(B_{msb}-1)} \\ &\quad - x_{0,M} - x_{0,L} 2^{-(B_{msb}-1)}) \\ &= f_M(x_M) + f_L(x_L) 2^{-(B_{msb}-1)} \end{aligned} \quad (26)$$

where

$$\begin{aligned} f_M(x_M) &= f(x_0) + f'(x_0)(x_M - x_{0,M}) \\ f_L(x_L) &= f'(x_0)(x_L - x_{0,L}) \end{aligned}$$

and $f_M(x_M)$ can be used as the **E**-block. Note that in the decomposition in (26), only x is decomposed into MSB and LSB components and the factor $f'(x_0)$ remains in full precision. If a simpler **E**-block is required, $f'(x_0)$ can also be decomposed into MSB and LSB part, as shown in Section III-D. The pivot point x_0 should be chosen such that the error metric, e.g., the mean square error, between the original and the E-ANT kernel is minimized.

C. DPD via piecewise linear (PWL) approximation

The PWL approximation can be employed when $f(x)$ ($x \in [x_l, x_u]$) is non-differentiable or the input dynamic range is large.

The PWL approximation employs $N + 1$ points $(x_k, f(x_k))$ where $x_k = x_l + \frac{k}{N}(x_u - x_l)$ and $k = 0, 1, \dots, N$ to approximate $f(x)$ as:

$$\begin{aligned} f(x) &\approx \sum_{k=1}^N p_k(x) \\ p_k(x) &= \begin{cases} a_k x + b_k & x_k \leq x < x_{k+1} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (27)$$

where $x_0 = x_l$, $x_N = x_u$, $a_k = \frac{f(x_{k+1}) - f(x_k)}{x_{k+1} - x_k}$, and $b_k = \frac{x_{k+1}f(x_k) - x_k f(x_{k+1})}{x_{k+1} - x_k}$. Each segment $p_k(x)$ can be decomposed by noting that for a linear function $p(x)$, substituting for x from (20), we have

$$\begin{aligned} p(x) &= p(x_M + x_L 2^{-(B_{msb}-1)}) \\ &= p(x_M) + p(x_L) 2^{-(B_{msb}-1)} \end{aligned} \quad (28)$$

Therefore, substituting (20) into (27), we obtain:

$$p_k(x) = \begin{cases} p_{k,M}(x_M) + p_{k,L}(x_L) 2^{-(B_{msb}-1)} & x_k \leq x < x_{k+1} \\ 0 & \text{otherwise} \end{cases} \quad (29)$$

where

$$\begin{aligned} p_{k,M}(x_M) &= a_k x_M + b_k \\ p_{k,L}(x_L) &= a_k x_L \end{aligned}$$

and $p_{k,M}(x_M)$ can be employed as the **E**-block.

Note that other piecewise approximation methods such as spline interpolation [16] where each segment is approximated with a low order polynomial can also be employed for DPD. Each low order polynomial can be decomposed in a manner similar to (24).

Next, we apply DPD to obtain E-ANT architectures for arithmetic units and compute kernels commonly used in signal processing and machine learning.

D. E-ANT Arithmetic Unit Architectures

1) *E-ANT Adder*: The output of a two-operand adder is given by:

$$y_a = x_1 + x_2$$

where x_1 and x_2 are the input operands. We first decompose the operands into MSB and LSB components according to (20):

$$\begin{aligned} x_1 &= x_{1M} + x_{1L} 2^{-(B_{msb}-1)} \\ x_2 &= x_{2M} + x_{2L} 2^{-(B_{msb}-1)} \end{aligned}$$

where x_{iM} and x_{iL} are defined in (21) and (22), respectively.

Since addition is a linear function, DPD can be easily obtained from (28) as follows:

$$\begin{aligned} y_a &= x_{1M} + x_{2M} + x_{1L} 2^{-(B_{msb}-1)} + x_{2L} 2^{-(B_{msb}-1)} \\ &= f_M + f_L 2^{-(B_{msb}-1)} \end{aligned} \quad (30)$$

where $f_M = x_{1M} + x_{2M}$ and $f_L = x_{1L} + x_{2L}$. The data flow graph (DFG) and the symbol of the E-ANT adder are shown in Fig. 3(a) and Fig. 3(b), respectively.

2) *E-ANT Multiplier*: Employing the DPD in (21)-(23), the E-ANT multiplier can be derived as follows:

$$\begin{aligned} y_a &= x_1 x_2 \\ &= (x_{1M} + x_{1L} 2^{-(B_{msb}-1)})(x_{2M} + x_{2L} 2^{-(B_{msb}-1)}) \\ &= f_M + f_L 2^{-(B_{msb}-1)} \end{aligned} \quad (31)$$

where $f_M = x_{1M} x_{2M}$ and $f_L = x_{1L} x_{2M} + x_{1M} x_{2L}$. Fig. 4(a) and Fig. 4(b) show the DFG and symbol of the E-ANT multiplier.

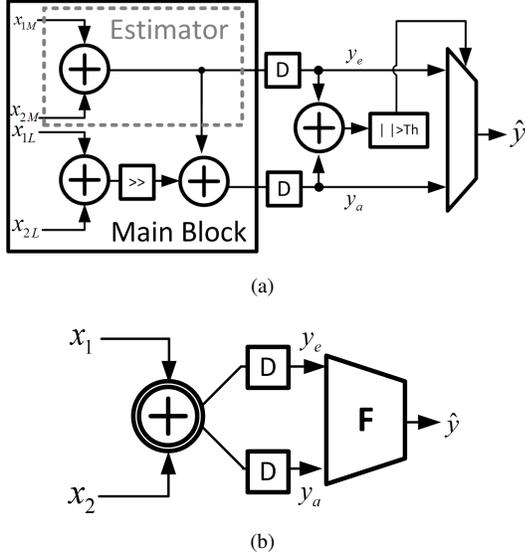


Figure 3. E-ANT Adder: a) DFG, and b) symbol.

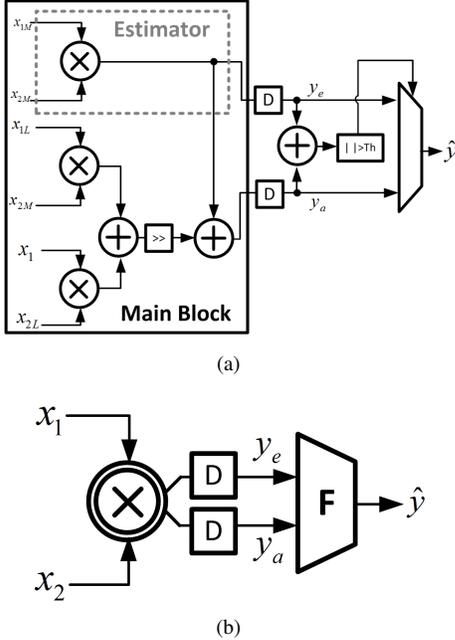


Figure 4. E-ANT Multiplier: a) DFG, and b) symbol.

3) *E-ANT multiply-accumulator (MAC)*: MAC operation is described as:

$$y_a[n] = x[n]w[n] + y_a[n-1] \quad (32)$$

We first decompose $x[n]$, $w[n]$ and $y[n-1]$ according to (20):

$$x[n] = x_M[n] + x_L[n]2^{-(B_{msb}-1)} \quad (33)$$

$$w[n] = w_M[n] + w_L[n]2^{-(B_{msb}-1)} \quad (34)$$

$$y_a[n-1] = y_{a,M}[n-1] + y_{a,L}[n-1]2^{-2(B_{msb}-1)} \quad (35)$$

The E-ANT MAC can be obtained by substituting (33)-(35) in to (32), and employing (30)-(31) to decompose $y_a[n]$ as follows:

$$\begin{aligned} y_a[n] &= x_M[n]w_M[n] + y_{a,M}[n-1] \\ &+ (x_L[n]w_M[n] + (x_M[n] + x_L[n]2^{-(B_{msb}-1)})w_L[n] \\ &+ y_{a,L}[n-1]2^{-(B_{msb}-1)})2^{-(B_{msb}-1)} \\ &= x_M[n]w_M[n] + y_{a,M}[n-1] \\ &+ (x_L[n]w_M[n] + x[n]w_L[n] \\ &+ y_{a,L}[n-1]2^{-(B_{msb}-1)})2^{-(B_{msb}-1)} \\ &= f_M + f_L2^{-(B_{msb}-1)} \\ &+ y_{a,L}[n-1]2^{-(B_{msb}-1)}2^{-(B_{msb}-1)} \end{aligned} \quad (36)$$

where $f_M = x_M[n]w_M[n] + y_{a,M}[n-1]$ and $f_L = x_L[n]w_M[n] + x[n]w_L[n] + y_{a,L}[n-1]2^{-(B_{msb}-1)}$. Fig. 5(a) and Fig. 5(b) show the DFG and the symbol of the E-ANT MAC.

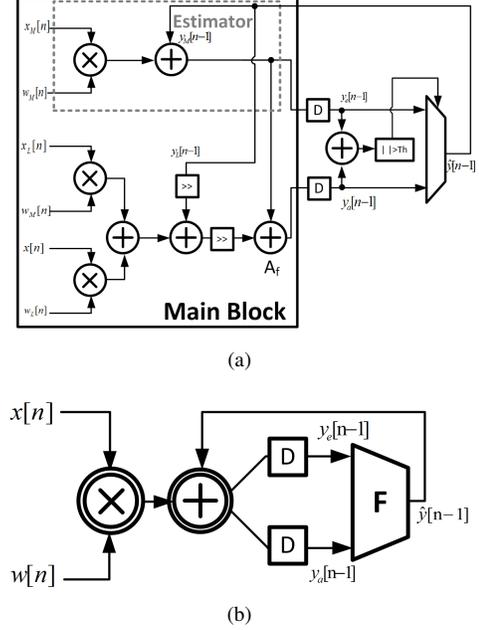


Figure 5. E-ANT MAC unit: a) DFG, and b) symbol.

E. E-ANT Signal Processing and Machine Learning Kernels

Complex E-ANT kernels can be derived by employing the E-ANT arithmetic units derived in section III-D.

1) *E-ANT FIR Filter*: One of the most important kernels in information processing is filtering/convolution. We can derive an E-ANT FIR filter by employing (30)-(31) as follows:

$$\begin{aligned} y_a[n] &= \sum_{i=0}^{N-1} w[i]x[n-i] \\ &= \sum_{i=0}^{N-1} f_{iM} + \sum_i f_{iL}2^{-(B_{msb}-1)} \end{aligned}$$

where $f_{iM} = x_M[n-i]w_M[i]$ and $f_{iL} = x_L[n-i]w_M[i] + x[n-i]w_L[i]$ for $i = 0 \dots N-1$. Fig. 6 shows the DFGs of the direct form and transposed form E-ANT FIR filter where we make use of the symbols in Fig. 3(b), 4(b), and 5(b) to simplify the DFGs.

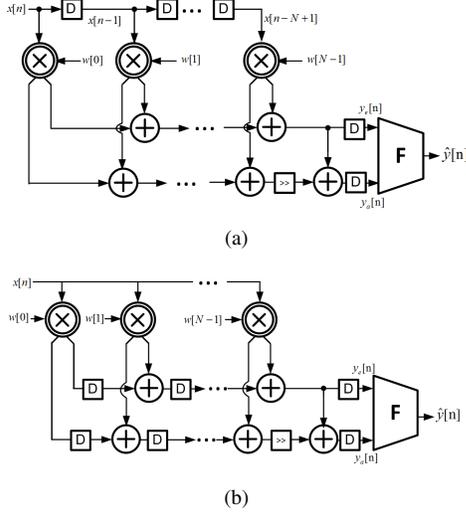


Figure 6. E-ANT FIR filter: a) the DFG of direct form FIR filter, and b) the DFG of transposed form FIR filter.

2) *E-ANT Fast Fourier Transform (FFT) Butterfly Unit (BU)*: BU is the main data processing unit in FFT processors. A general BU implements the following function:

$$\begin{aligned} y_{1r,a} &= x_{1r} + x_{2r} \\ y_{1i,a} &= x_{1i} + x_{2i} \\ d &= x_1 - x_2 \\ y_{2r,a} &= d_r W_r - d_i W_i \\ y_{2i,a} &= d_r W_i + d_i W_r \end{aligned}$$

where x_1 and x_2 are the inputs, y_1 and y_2 are the outputs, and W is the twiddle factor. The real and imaginary parts are denoted by r and i subscripts, respectively. E-ANT FFT BU can be derived as shown in Table I.

Table I
DPD FOR FFT BU

$y_{1r,a} = x_{1r} + x_{2r} = y_{1r,M} + y_{1r,L}2^{-(B_{msb}-1)}$, where $y_{1r,M} = x_{1r,M} + x_{2r,M}$ and $y_{1r,L} = x_{1r,L} + x_{2r,L}$.
$y_{1i,a} = x_{1i} + x_{2i} = y_{1i,M} + y_{1i,L}2^{-(B_{msb}-1)}$, where $y_{1i,M} = x_{1i,M} + x_{2i,M}$ and $y_{1i,L} = x_{1i,L} + x_{2i,L}$.
$y_{2r,a} = d_r W_r - d_i W_i = y_{2r,M} + y_{2r,L}2^{-(B_{msb}-1)}$, where $y_{2r,M} = d_{r,M} W_{r,M} - d_{i,M} W_{i,M}$ and $y_{2r,L} = d_{r,L} W_{r,M} + d_r W_{r,L} - (d_{i,L} W_{i,M} + d_i W_{i,L})$.
$y_{2i,a} = d_r W_i + d_i W_r = y_{2i,M} + y_{2i,L}2^{-(B_{msb}-1)}$, where $y_{2i,M} = d_{r,M} W_{i,M} + d_{i,M} W_{r,M}$ and $y_{2i,L} = d_{r,L} W_{i,M} + d_r W_{i,L} + d_{i,L} W_{r,M} + d_i W_{r,L}$.

The DFG of the E-ANT FFT BU is shown in Fig. 7.

3) *E-ANT Exponential Kernel*: Exponential kernel (e^{-x}) is a critical component in many machine learning algorithms such as kernel SVM [12], [17], Gaussian Mixture Model [18], and others [19], [20]. Taylor expansion in Section III-B and

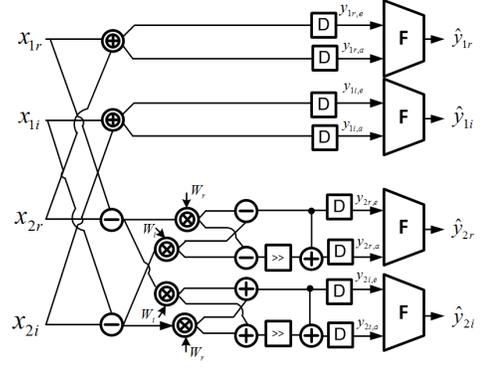


Figure 7. DFG of the E-ANT FFT butterfly unit.

PWL approximation in Section III-C can be employed to obtain E-ANT exponential kernels.

Assuming that the input dynamic range is scaled to $[0,1]$, a 2^{nd} order Taylor expansion leads to:

$$y_a = e^{-x} \approx e^{-x_0} - e^{-x_0} \times (x - x_0) + \frac{e^{-x_0}(x - x_0)^2}{2} \quad (37)$$

When $x_0 = 0.5$, (37) simplifies to

$$y_a \approx ax^2 + bx + c \quad (38)$$

where $a = 0.3033$, $b = -0.9098$, and $c = 0.9856$. The Taylor expansion based E-ANT exponential block can thus be derived from (24) as follows:

$$y_a \approx f_M + f_L 2^{-(B_{msb}-1)}$$

where

$$\begin{aligned} f_M &= ax_M^2 + bx_M + c \\ f_L &= a(2x_M x_L + x_L^2 2^{-(B_{msb}-1)}) + bx_L \end{aligned}$$

The DFG is shown in Fig. 8(a).

Alternatively, PWL approximation can be employed to obtain an E-ANT exponential kernel. Assume that two linear functions on $[0, 0.5]$ and $[0.5, 1]$ are used to approximate the exponential function on the interval $[0, 1]$, then according to (27):

$$y_a = e^{-x} \approx \begin{cases} a_1 x + b_1 & 0 \leq x < 0.5 \\ a_2 x + b_2 & 0.5 \leq x < 1 \end{cases}$$

where $a_1 = -0.7869$, $b_1 = 1$, $a_2 = -0.4773$ and $b_2 = 0.8452$. We first decompose a_i , b_i ($i = 1, 2$) according to (20):

$$\begin{aligned} a_i &= a_{i,M} + a_{i,L} 2^{-(B_{msb}-1)} \\ b_i &= b_{i,M} + b_{i,L} 2^{-2(B_{msb}-1)} \end{aligned}$$

Since each segment is linear, they can be decomposed by using (28):

$$\begin{aligned} y_{a,i} &= a_i x + b_i \\ &= f_{i,M} + f_{i,L} 2^{-(B_{msb}-1)} \end{aligned}$$

where

$$\begin{aligned} f_{i,M} &= a_{i,M} x_M + b_{i,M} \\ f_{i,L} &= a_{i,L} x_L + a_{i,L} x + b_{i,L} 2^{-(B_{msb}-1)} \end{aligned}$$

The resultant E-ANT exponential kernel has a reconfigurable architecture where different approximations are chosen according to the input values. The DFG of the PWL based E-ANT exponential kernel is shown in Fig. 8(b).

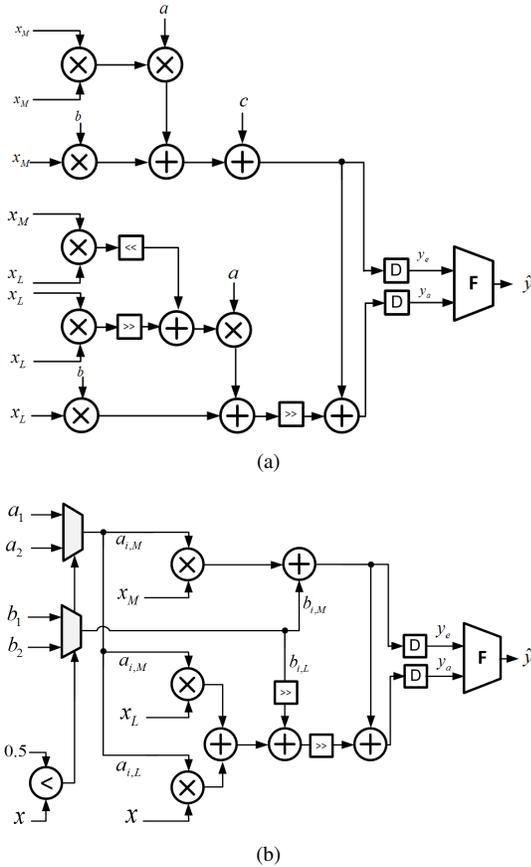


Figure 8. DFG of the E-ANT exponential kernel: a) Taylor expansion based, and b) PWL approximation based

IV. SIMULATION RESULTS

This section presents the design optimization of the proposed E-ANT technique, and shows the simulation results in a 45 nm CMOS process when E-ANT is applied to an SVM EEG classification system. This system is chosen due to its data path dominant architecture, which makes it an attractive platform to apply DPD. It has been shown that other machine learning methods such as extreme learning machines (ELMs) [21], [22], [23] might outperform SVM in terms of learning speed or accuracy in EEG seizure classification problems. DPD is also applicable to these algorithms. However, in this paper, we focus on SVM as it has been implemented in integrated circuits [24] and hence offers a point of comparison.

A. Evaluation Methodology

Figure 9(a) shows the evaluation methodology employed to quantify system-level performance metrics and to estimate system-level energy consumption that integrates circuit, architecture, and system level design variables. The methodology consists of two parts: 1) *system-level error injection*, and 2) *system-level energy estimation*. Comparison of the proposed E-ANT with conventional approach (no error compensation) and retraining based approach in [11] is done using a commercial 45 nm CMOS process. The results are shown in Section IV-E in Fig. 14.

System-level error injection is done as follows:

- 1) Characterize delay vs. V_{dd} of basic gates such as AND and XOR using HSPICE for $0.2 \text{ V} \leq V_{dd} \leq 1.2 \text{ V}$.
- 2) Develop structural Verilog HDL models of key kernels needed in the EEG classification system using the basic gates characterized in Step 1. These kernels are a 12 b input, 8 b coefficient, and 16 b output, 44-tap FIR filter (used in the FE) and a 8 b input, 8 b coefficient, and 19 b output vector-matrix multiplication kernel (used in the SVM CE). In designing the FE and CE, floating point models were first developed to verify functionality. To convert floating point design to fixed point data path, we instrumented the dynamic range at the output of the FIR filter and the vector-matrix multiplication kernel using application level data. The filter output was truncated to 16 b with 1 b sign bit, 2 b integer length, and 13 b fraction length. Full precision was maintained in the absolute-sum block as shown in Fig. 12, and the output was truncated and normalized to 8 b with 1 b sign bit, and 7 b fraction as required by the SVM algorithm. Truncation was performed at the output of the first and second MAC in the CE (see Fig. 12) to manage bit-growth in the similar manner.
- 3) HDL simulations of these kernels were conducted at different voltages by including the appropriate voltage-specific delay numbers obtained in Step 1 into the HDL model. The error PMFs of these kernels and error rates p_η are obtained for different supply voltages (and thus voltage overscaling factor K_{vos}).
- 4) System performance evaluation and design optimization are done by injecting errors into a fixed point MATLAB-model of the EEG classification system. The errors are obtained by sampling the error PMFs obtained in Step 3.

Figure 9(c) shows the error PMF $P_\eta(\eta)$ of the 44-tap low pass FIR filter used in the FE at $V_{dd} = 0.9 \text{ V}$ ($f_{clk} = 76 \text{ MHz}$) which corresponds to a $K_{vos} = 0.75$ and an error rate $p_\eta = 0.05$, and Fig. 9(d) shows the error rate p_η increases from 10^{-5} at $V_{dd} = 1.15 \text{ V}$ to 0.99 at $V_{dd} = 0.5 \text{ V}$ as the voltage scales down.

System-level energy estimation is done as follows:

- 1) Obtain a full adder (FA) count N_{FA} of the kernel being analyzed.
- 2) Conduct a one-time characterization of the energy consumption of a FA incorporating both dynamic and leakage energies as follows:

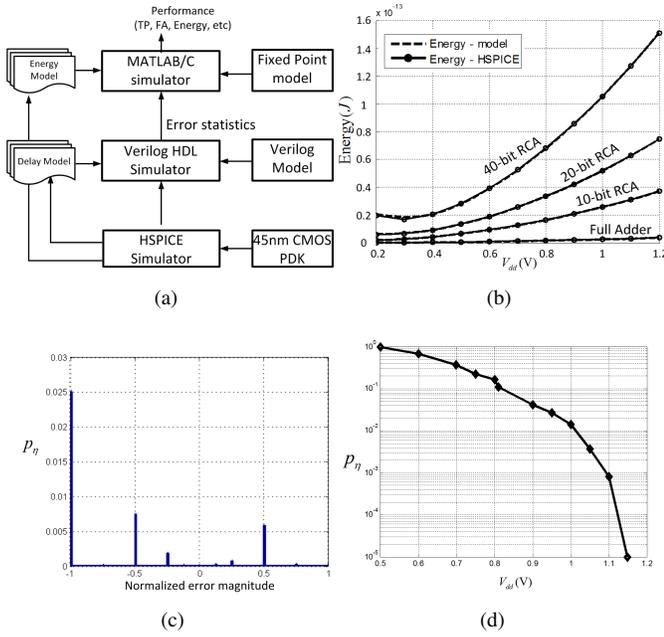


Figure 9. Evaluation methodology: a) simulation setup, and b) comparison of the energy model and HSPICE simulations in a 45 nm CMOS process, c) error PMF at $V_{dd} = 0.9$ V, and d) error rate p_η vs. V_{dd} for the 44-tap low pass filter employed in the FE, the CHB-MIT EEG data set [11] is employed as input.

$$E_{FA} = C_{FA}V_{dd}^2 + V_{dd}I_{leak}(V_{dd})\frac{1}{f_{clk}} \quad (39)$$

with

$$I_{leak}(V_{dd}) = \mu C_{ox} \frac{W}{L} (m-1) V_T^2 e^{\frac{-V_t}{mV_T}} e^{\frac{-\eta_d V_{dd}}{mV_T}} (1 - e^{\frac{-V_{dd}}{V_T}}) \quad (40)$$

where C_{FA} is the effective load capacitance of the FA and is extracted from HSPICE, V_{dd} is the supply voltage, V_t , V_T , μ , C_{ox} , and η_d are the threshold voltage, the thermal voltage, the carrier mobility, the gate capacitance per unit W/L , and the drain induced barrier lowering (DIBL) coefficient, respectively, obtained from the process files, and m is a constant related to the sub-threshold slope factor and is a fitting parameter.

- 3) The energy estimate of the kernel is obtained as $E_{op} = N_{FA}E_{FA}$.

Figure 9(b) shows the modeling results of the FA and ripple carry adder (RCA) for various bit width demonstrating the accuracy and scalability of the energy model. The energy model is within 5% (for 0.2 V $\leq V_{dd} \leq 1.2$ V) of circuit simulation results.

B. E-ANT Design Optimization

The methodology in Fig. 9(a) is employed to perform optimization for the E-ANT kernels proposed in Sect. III, and the E-ANT MAC kernel in Fig. 5(a) is used as an example. Since we adopt VOS to obtain different error rates, the parameters to be optimized are the voltage overscaling factor K_{vos} and the E-block bit width B_{msb} , where we assume that $B_{x,msb} = B_{w,msb} = B_{msb}$. The optimization framework

is general enough to include the case when $B_{x,msb} \neq B_{w,msb}$. A grid search algorithm is employed to systematically determine the optimum setting K_{vos}^* and B_{msb}^* satisfying the performance metric, as shown in Algorithm 1 below. We adopt mean squared error (MSE) with respect to the floating point kernel as the performance metric:

$$MSE = E(\hat{y} - y_{fl})^2 \quad (41)$$

where \hat{y} and y_{fl} indicate the E-ANT and floating point output, respectively. The maximum E-block length B_{max} under which the E-block does not make errors is determined by K_{vos} . The optimization routine gives the optimum E-ANT configuration, including K_{vos}^* , B_{msb}^* and minimum energy E_{op}^* , at the output.

Algorithm 1 Energy optimization algorithm for E-ANT

1. Initialize $K_{vos}^* = 1$, $B_{msb}^* = 0$, $E_{op}^* =$ energy of conventional MAC, $MSE_{req} =$ specified MSE requirement.
2. $K_{vos} = K_{vos} - \Delta$, $B_{msb} = 0$. Obtain maximum E-block precision B_{max} to ensure error-free E-block operation.
3. $B_{msb} = B_{msb} + 1$. If $B_{msb} > B_{max}$, then exit, else compute MSE according to (23).
4. If $MSE < MSE_{req}$, then calculate energy $E(K_{vos})$ according to (21), else go to step 3
5. If $E_{op}^* > E(K_{vos})$, then $E_{op}^* = E(K_{vos})$, and $B_{msb}^* = B_{msb}$
6. Go to step 2

Algorithm 1 is employed to optimize E-ANT MAC for 8b and 16b precision with MSE requirements of $10^{-2} \sim 10^{-5}$. The iso-MSE plots in the B_{msb} and p_η plane (see Fig. 10(a) and Fig. 10(c)) indicate that the optimum B_{msb} increases as the error rate p_η increases because a higher precision E-block is needed to compensate for the M-block errors. Figure 10(b) shows that the 8 bit E-ANT MAC achieves energy savings of 16% \sim 69%, while as the 8 bit ANT MAC fails to achieve energy savings at the tight MSE requirement of 10^{-5} due to E-block overheads. Figure 10(d) shows that the 16 bit E-ANT MAC achieves 59% energy savings compared with the conventional MAC. The overhead of the E-ANT architecture is below 8% compared with the 36.4% and 13.9% overheads for the 8 bit and 16 bit ANT architecture, respectively.

Figure 11 shows that the energy savings increase as the MSE requirement increases for a fixed B_x . This is because a larger MSE requirement allows the MAC to operate at a higher p_η and can thus reduce the E-block overheads. This is also confirmed in Fig. 10(b) and Fig. 10(d). Additionally, the energy savings increase as B_x increases for a fixed MSE requirement because a large B_x tends to tolerate more LSB errors, thus enabling the MAC to operate at a higher p_η .

C. EEG Classification System Set-up

To evaluate the performance of E-ANT, the filter kernel in the FE and the vector-matrix multiplication kernel in the SVM CE shown in Fig. 12 are implemented employing the E-ANT MAC as shown in Fig. 5, and are characterized via the procedure described in section IV-B. For the filter bank in

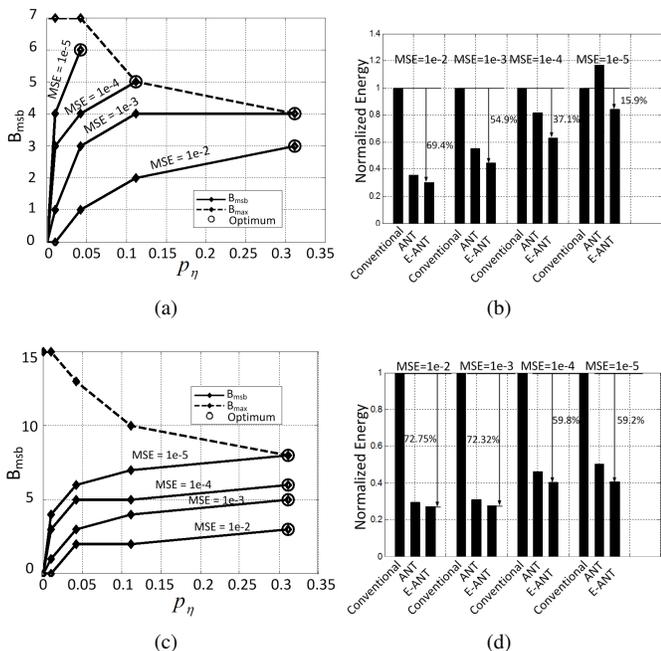


Figure 10. Optimization of E-ANT MAC, the dashed line illustrates that the maximum E-block precision B_{max} decreases as p_η increases, indicating that to ensure error-free E-block operation, the E-block bit width is upper bounded. The solid lines show the optimum B_{msb} configuration for each p_η at different MSE requirements, with the circle marker indicating the (B_{msb}^*, p_η^*) pair achieving the MSE requirements with minimum E_{op} : a) optimization results of an 8×8 E-ANT MAC for different MSE requirements, b) normalized energy of an 8×8 conventional MAC, ANT MAC and E-ANT MAC, c) optimization results of a 16×16 E-ANT MAC for different MSE requirements, and d) normalized energy of a 16×16 conventional MAC, ANT MAC, and E-ANT MAC.

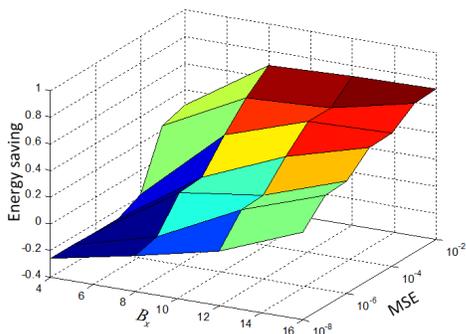


Figure 11. Energy savings vs. input precision and MSE.

the FE, we use an input of 12 bit, with the MSB 8 bit taken as the E-block. For the CE, the input precisions of the two MACs are chosen to be 8 bit, and E-block precisions are 4 bit.

We employ the CHB-MIT EEG data set [11] to train the SVM and use leave-one-out cross validations to evaluate the system performance. The system performance metric employed is the true positive (TP) rate p_{tp} and false positive/alarm (FP) rate p_{fpp} , defined as:

$$p_{tp} = \frac{TP}{TP + FN}$$

$$p_{fpp} = \frac{FP}{FP + TN}$$

where TP , FN , FP , and TN are the number of true positives, false negatives, false positives, and true negatives, respectively. A good classifier achieves high values of p_{tp} (> 0.9) at a small constant false alarm rate p_{fpp} (< 0.01).

Three implementations are considered, i.e., the uncompensated system (denoted as CONV), the system which performs retraining with erroneous features, similar to the one proposed in [11] (denoted as RETRAIN), and the system with E-ANT (denoted as E-ANT). In the retraining method [11], the classifier is trained with features extracted in the presence of VOS errors. Unlike in the retraining method [11] where the CE needs to be error free, E-ANT can tolerate errors in both the FE and CE. Therefore, two setups are considered in our experiment: 1) errors in FE only, and 2) errors in both FE and CE. The maximum value of the error rate p_η for which $p_{tp} > 0.9$ and $p_{fpp} < 0.01$ is referred to as the error tolerance $p_{\eta-max}$ of the architecture. In the first setup, $p_{\eta-max}$ is the error rate in the FE, and in the second setup, $p_{\eta-max}$ is the maximum of the error rate in the FE and CE.

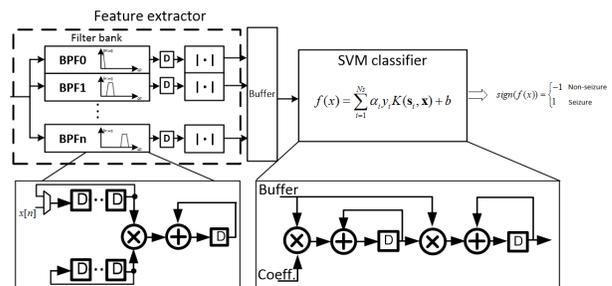


Figure 12. Second order polynomial kernel SVM EEG classification system architecture.

D. SNR Performance

Figure 13(a) shows that the improvement in the application level SNR (see (7)-(9)) achieved by E-ANT at the FE output is significant. In particular, the SNR of the M-block (also the SNR of the conventional system with errors), $SNR_{M,a}$, drops catastrophically from 42 dB to 10 dB for values of p_η as low as 8×10^{-4} . The SNR of the E-block, $SNR_{E,a}$, is constant at 23 dB for $p_\eta \leq 0.42$. This is because the E-block makes small magnitude estimation errors e . For $p_\eta > 0.42$, $SNR_{E,a}$ drops catastrophically as the E-block also starts to make large magnitude timing errors. In contrast, the ANT system SNR, $SNR_{ANT,a}$, is at least 10 dB higher than either $SNR_{E,a}$ or $SNR_{M,a}$ for values of p_η as high as 0.1, and approaches the E-block SNR as p_η increases.

The circuit level SNRs (see (10) - (12)) also exhibit a similar trend in Fig. 13(b). Furthermore, the SNR analysis in section II-A is validated by plotting (13) and (14) in Fig. 13(a) and Fig. 13(b), respectively.

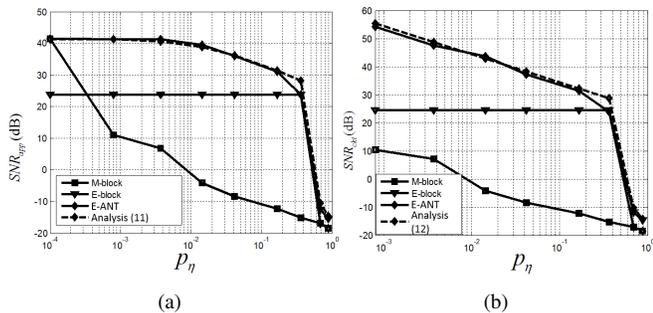


Figure 13. SNR at the output of the FE: a) application level SNR, and b) circuit level SNR.

E. Classification Performance and Energy Savings

As shown in Fig. 14(a), p_{tp} drops sharply as circuit error rate increases in CONV system where no SEC is applied. The RETRAIN system does slightly better than the CONV system because the classifier is retrained to adapt to the error affected features. However, $p_{\eta-max}$ is still only around 10^{-3} . This is due to the fact that the errors under investigation are timing errors. Unlike the stuck-at-faults errors in [11], timing errors are dynamic and depend on the state of circuit, so the error pattern observed during training might not be the same as during the test. In contrast, when E-ANT is applied, p_{tp} degrades gracefully as p_{η} increases. As a result, the E-ANT system can achieve $p_{\eta-max}$ as high as 0.38. Figure 14(a) also shows that the p_{tp} is always lower than 0.9 when only E-block is employed, i.e., the E-block on its own is unable to meet the performance specifications. Similarly, when errors present in both FE and CE (Fig. 14(b)), both the CONV system and RETRAIN system achieve $p_{\eta-max} < 10^{-3}$, while E-ANT achieves a $p_{\eta-max}$ of 0.17. These are of 2-orders-of-magnitude (errors in FE only) and 3-orders-of-magnitude (errors in both FE and CE) greater than the existing systems. The receiver operating characteristic (ROC) curve at $p_{\eta-max}$ is shown in Fig. 14(c) when errors are in FE only, and in Fig. 14(d) when errors are in both FE and CE. In both experiments, the ROC of the CONV as well as the RETRAIN system approaches the ROC of a random classifier which outputs ± 1 with equal probability, while the ROC of the E-ANT system (w/ or w/o retraining) remains close to the ROC of an ideal classifier.

Principle component analysis (PCA) is performed on the feature vectors to understand the reason why CONV system fails but E-ANT system is able to maintain good performance. Fig. 14(e) shows that when no SEC is applied, circuit errors have two effects on the feature vectors: 1) errors make it harder to separate the positive and negative samples, and 2) the entire feature space is shifted due to the accumulation block in the FE. The SVM fails to correctly perform classification without knowledge of the error statistics. Figure 14(f) shows that the large magnitude errors are compensated and converted to small residual errors when E-ANT is applied. This will cause a very small shift in the feature space. As a result, the SVM classifier can still perform correct classification. One way to improve E-ANT further is to incorporate retraining. In this method,

the classifier is trained employing features that are subject to residual errors after the correction via E-ANT. However, as shown in Fig. 14(a), the improvement is minor due to the fact that the residual errors are typically small.

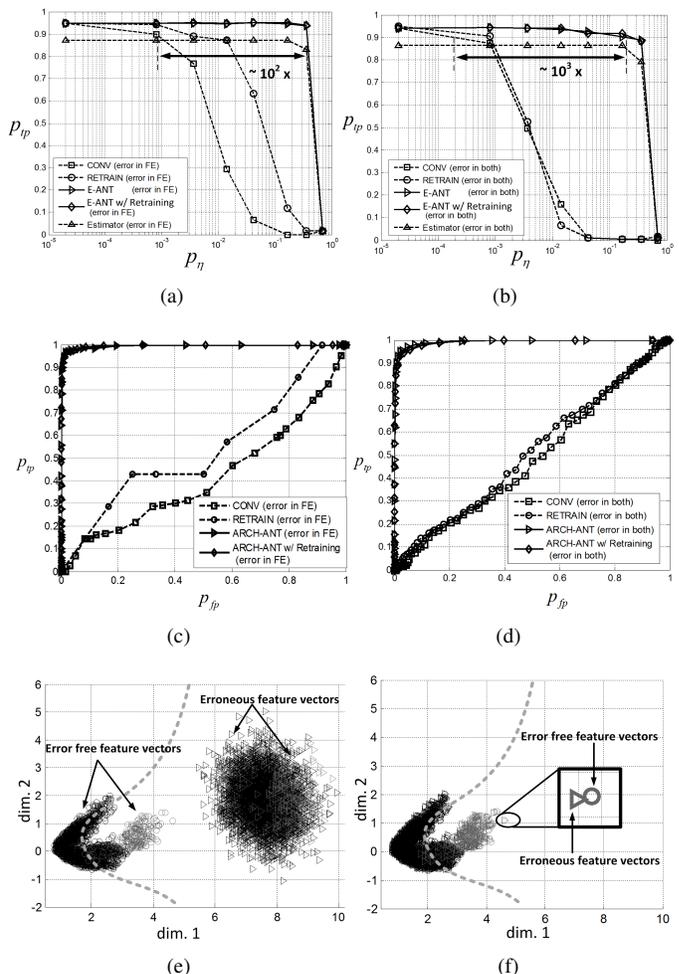


Figure 14. Simulation results: a) p_{tp} of CONV, RETRAIN and E-ANT with $p_{ffp} = 0.01$ when errors are in FE only, b) p_{tp} of CONV, RETRAIN and E-ANT with $p_{ffp} = 0.01$ when errors are in both FE and CE, c) ROC curve of CONV, RETRAIN and E-ANT at $p_{\eta-max}$ when errors are in FE only, d) ROC curve of CONV, RETRAIN and E-ANT at $p_{\eta-max}$ when errors are in both FE and CE, e) PCA results of error-free and erroneous features for CONV, and f) PCA results of error-free and erroneous features for E-ANT.

Table II compares the $p_{\eta-max}$, FE energy/feature (E_F), and CE energy/decision (E_C) of the three systems. The E-ANT system can achieve $p_{\eta-max}$ of 0.38 when errors are in FE only, and 0.17 when errors are in both FE and CE. When VOS is applied for energy saving, the E-ANT system is able to achieve 51% energy savings when errors are in FE only compared with the CONV system. When both FE and CE are in error, the E-ANT system is able to achieve 43% and 29% energy savings in the FE and CE, respectively.

V. CONCLUSIONS

In this paper, an embedded statistical error compensation technique named E-ANT is proposed and employed as a low power technique in designing machine learning and signal

Table II
PERFORMANCE AND ENERGY COMPARISON

	Errors in FE only		Errors in both FE and CE		
	$p_{\eta-max}$	Energy savings in FE	$p_{\eta-max}$	Energy savings in FE	Energy savings in CE
Conventional	8×10^{-4}	NA	2×10^{-4}	NA	NA
Error resilient retraining	3×10^{-3}	7%	8×10^{-4}	13%	12%
E-ANT	0.38	51%	0.17	43%	29%

processing kernels. E-ANT employs DPD to embed the RPR E-block into the M-block. The DPD is general and is derived for a wide class of compute kernels. The effectiveness of the proposed E-ANT technique has been demonstrated through the design of an SVM EEG seizure classification system where simulation results in a commercial 45 nm CMOS process show that E-ANT achieves up to 0.38 error tolerance and up to 51% energy savings compared with an uncompensated system.

This work shows that by exploring architectural transforms, it is possible to design architectures that are inherently error resilient without an explicit E-block. It opens up a few research directions to extend or generalize existing SEC techniques. In particular, other low complexity error compensation techniques can be investigated that improve upon existing SEC techniques such as SSNOC and soft-NMR [2]. Moreover, with the adoption near/sub-threshold voltage design and continuous scaling of CMOS process, PVT induced errors and defects induced errors are becoming a growing concern for the design of low power embedded platforms. The effectiveness of error resiliency techniques to enhance system robustness in presence of these new error models can also be explored.

ACKNOWLEDGMENT

This work was supported in part by Systems on Nanoscale Information fabriCs (SONIC), one of the six SRC STARnet Centers, sponsored by MARCO and DARPA.

APPENDIX - A

In this appendix, we derive (24). Consider the n th order Taylor expansion of $f(x)$ as shown below:

$$f(x) = f(x_0) + \sum_{k=1}^n \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k \quad (42)$$

Substituting the binomial expansion of $(x - x_0)^k$ in (42), we obtain

$$\begin{aligned} f(x) &= f(x_0) \\ &+ \sum_{k=1}^n \frac{f^{(k)}(x_0)}{k!} \sum_{i=0}^k \binom{k}{i} x^i (-x_0)^{k-i} \\ &= f(x_0) \\ &+ \sum_{k=1}^n \sum_{i=0}^k \left[\frac{f^{(k)}(x_0)}{k!} \binom{k}{i} (-x_0)^{k-i} \right] x^i \quad (43) \end{aligned}$$

Next, employing (20) and the binomial expansion, we write x^i as:

$$\begin{aligned} x^i &= (x_M + x_L 2^{-(B_{msb}-1)})^i \\ &= \sum_{j=0}^i \binom{i}{j} x_M^j (x_L 2^{-(B_{msb}-1)})^{i-j} \\ &= x_M^i + \sum_{j=0}^{i-1} \binom{i}{j} x_M^j (x_L 2^{-(B_{msb}-1)})^{i-j} \\ &= x_{i,M} + x_{i,L} \quad (44) \end{aligned}$$

where

$$\begin{aligned} x_{i,M} &= x_M^i \\ x_{i,L} &= \sum_{j=0}^{i-1} \binom{i}{j} x_M^j (x_L 2^{-(B_{msb}-1)})^{i-j} \end{aligned}$$

Substituting (44) into (43), we obtain:

$$\begin{aligned} f(x) &\approx f(x_0) \\ &+ \sum_{k=1}^n \sum_{i=0}^k \left[\frac{f^{(k)}(x_0)}{k!} \binom{k}{i} (-x_0)^{k-i} \right] x^i \\ &= f(x_0) \\ &+ \sum_{k=1}^n \sum_{i=0}^k \left[\frac{f^{(k)}(x_0)}{k!} \binom{k}{i} (-x_0)^{k-i} \right] (x_{i,M} + x_{i,L}) \\ &= f_M(x_M) + f_L(x_M, x_L) \quad (45) \end{aligned}$$

where

$$\begin{aligned} f_M(x_M) &= f(x_0) \\ &+ \sum_{k=1}^n \sum_{i=0}^k \left[\frac{f^{(k)}(x_0)}{k!} \binom{k}{i} (-x_0)^{k-i} \right] x_{i,M} \\ f_L(x_M, x_L) &= \sum_{k=1}^n \sum_{i=0}^k \left[\frac{f^{(k)}(x_0)}{k!} \binom{k}{i} (-x_0)^{k-i} \right] x_{i,L} \end{aligned}$$

This completes the derivation of (24).

REFERENCES

- [1] S. Zhang and N. Shanbhag, "Embedded error compensation for energy efficient dsp systems," in *Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on*, Dec 2014, pp. 30-34.
- [2] N. Shanbhag, R. Abdallah, R. Kumar, and D. Jones, "Stochastic computation," in *Design Automation Conference (DAC), 2010 47th ACM/IEEE*, June 2010, pp. 859-864.
- [3] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variations and impact on circuits and microarchitecture," in *Design Automation Conference, 2003. Proceedings*, June 2003, pp. 338-342.

- [4] K. Nepal, R. Bahar, J. Mundy, W. Patterson, and A. Zaslavsky, "Techniques for designing noise-tolerant multi-level combinational circuits," in *Design, Automation Test in Europe Conference Exhibition, 2007. DATE '07*, April 2007, pp. 1–6.
- [5] D. Ernst, S. Das, S. Lee, D. Blaauw, T. Austin, T. Mudge, N. S. Kim, and K. Flautner, "RAZOR: circuit-level correction of timing errors for low-power operation," *Micro, IEEE*, vol. 24, no. 6, pp. 10–20, Nov 2004.
- [6] D. Blaauw, S. Kalaiselvan, K. Lai, W.-H. Ma, S. Pant, C. Tokunaga, S. Das, and D. Bull, "RAZOR II: In situ error detection and correction for PVT and SER tolerance," in *Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International*, Feb 2008, pp. 400–622.
- [7] J. Tschanz, K. Bowman, C. Wilkerson, S.-L. Lu, and T. Karnik, "Resilient circuits; enabling energy-efficient performance and reliability," in *Computer-Aided Design - Digest of Technical Papers, 2009. ICCAD 2009. IEEE/ACM International Conference on*, Nov 2009, pp. 71–73.
- [8] C.-H. Chen, D. Blaauw, D. Sylvester, and Z. Zhang, "Design and evaluation of confidence-driven error-resilient systems," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 22, no. 8, pp. 1727–1737, Aug 2014.
- [9] R. Abdallah and N. Shanbhag, "An Energy-Efficient ECG Processor in 45-nm CMOS Using Statistical Error Compensation," *Solid-State Circuits, IEEE Journal of*, vol. 48, no. 11, pp. 2882–2893, Nov 2013.
- [10] M. Chen, J. Han, Y. Zhang, Y. Zou, Y. Li, and X. Zeng, "An error-resilient wavelet-based eeg processor under voltage overscaling," in *Biomedical Circuits and Systems Conference (BioCAS), 2014 IEEE*, Oct 2014, pp. 628–631.
- [11] N. Verma, K. H. Lee, K. J. Jang, and A. Shoeb, "Enabling system-level platform resilience through embedded data-driven inference capabilities in electronic devices," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, March 2012, pp. 5285–5288.
- [12] J. Yoo, L. Yan, D. El-Damak, M. Altaf, A. Shoeb, and A. Chandrakasan, "An 8-Channel Scalable EEG Acquisition SoC With Patient-Specific Seizure Classification and Recording Processor," *Solid-State Circuits, IEEE Journal of*, vol. 48, no. 1, pp. 214–228, Jan 2013.
- [13] V. Vapnik, "An overview of statistical learning theory," *Neural Networks, IEEE Transactions on*, vol. 10, no. 5, pp. 988–999, Sep 1999.
- [14] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.
- [15] K. K. Parhi, *VLSI digital signal processing systems : design and implementation*. New York: Wiley, 1999, a Wiley-Interscience publication. [Online]. Available: <http://opac.inria.fr/record=b1097682>
- [16] M. T. Heath, *Scientific Computing: An Introductory Survey*, 2nd ed., E. M. Munson, Ed. McGraw-Hill Higher Education, 1996.
- [17] A. Ganapathiraju, J. Hamaker, and J. Picone, "Applications of Support Vector Machines to speech recognition," *Signal Processing, IEEE Transactions on*, vol. 52, no. 8, pp. 2348–2355, Aug 2004.
- [18] H. Hu, M.-X. Xu, and W. Wu, "GMM Supervector Based SVM with Spectral Features for Speech Emotion Recognition," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4, April 2007, pp. IV-413–IV-416.
- [19] S. Lawrence, C. Giles, A. C. Tsoi, and A. Back, "Face recognition: a convolutional neural-network approach," *Neural Networks, IEEE Transactions on*, vol. 8, no. 1, pp. 98–113, Jan 1997.
- [20] D. Wedge, D. Ingram, D. McLean, and Z. Bandar, "On global-local artificial neural networks for function approximation," *Neural Networks, IEEE Transactions on*, vol. 17, no. 4, pp. 942–952, July 2006.
- [21] Y. Song, J. Crowcroft, and J. Zhang, "Automatic epileptic seizure detection in {EEGs} based on optimized sample entropy and extreme learning machine," *Journal of Neuroscience Methods*, vol. 210, no. 2, pp. 132 – 146, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0165027012002671>
- [22] Q. Yuan, W. Zhou, S. Li, and D. Cai, "Epileptic {EEG} classification based on extreme learning machine and nonlinear features," *Epilepsy Research*, vol. 96, no. 1&A52, pp. 29 – 38, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0920121111001185>
- [23] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 2, pp. 513–529, April 2012.
- [24] K. Lee, S.-Y. Kung, and N. Verma, "Low-energy formulations of support vector machine kernel functions for biomedical sensor applications," *Journal of Signal Processing Systems*, vol. 69, no. 3, pp. 339–349, 2012.



Sai Zhang received the B.S. in Electronic Science and Technology (with distinction) from University of Science and Technology of China. Since 2010, he has joined the University of Illinois at Urbana-Champaign and received the M.S. degree in Electrical and Computer Engineering in 2012. Currently, he is a PhD student in Electrical and Computer Engineering, University of Illinois at Urbana-Champaign. Since 2010, he has been a Research Assistant in Co-ordinated Science Laboratory. During the summers of 2013 and 2014, he was with Texas Instruments, Inc., Dallas, in the Embedded Processing Systems R&D Labs, where he was involved with designing reconfigurable error correction codec for memory and researching written once memory coding schemes for NVMs. His research interests are in the design of low power error resilient integrated circuits and systems for digital signal processing and machine learning. He is a student member of the IEEE.



Naresh R. Shanbhag is the Jack Kilby Professor of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign. He received his Ph.D. degree from the University of Minnesota (1993) in Electrical Engineering. From 1993 to 1995, he worked at AT&T Bell Laboratories at Murray Hill where he led the design of high-speed transceiver chip-sets for very high-speed digital subscriber line (VDSL), before joining the University of Illinois at Urbana-Champaign in August 1995. He has held visiting faculty appointments at the National Taiwan University (Aug.-Dec. 2007) and Stanford University (Aug.-Dec. 2014). His research interests are in the design of energy-efficient integrated circuits and systems for communications, signal processing and machine learning. He has more than 200 publications in this area and holds thirteen US patents. He is also a co-author of the research monograph *Pipelined Adaptive Digital Filters* published by Kluwer Academic Publishers in 1994. Dr. Shanbhag received the 2010 Richard Newton GSRC Industrial Impact Award, became an IEEE Fellow in 2006, received the 2006 IEEE Journal of Solid-State Circuits Best Paper Award, the 2001 IEEE Transactions on VLSI Best Paper Award, the 1999 IEEE Leon K. Kirchmayer Best Paper Award, the 1999 Xerox Faculty Award, the IEEE Circuits and Systems Society Distinguished Lecturership in 1997, the National Science Foundation CAREER Award in 1996, and the 1994 Darlington Best Paper Award from the IEEE Circuits and Systems Society. In 2000, Dr. Shanbhag co-founded and served as the Chief Technology Officer of Intersymbol Communications, Inc., (acquired in 2007 by Finisar Corp (NASDAQ:FNSR)) a semiconductor start-up that provided DSP-enhanced mixed-signal ICs for electronic dispersion compensation of OC-192 optical links. Since January 2013, he is the founding Director of the Systems On Nanoscale Information fabriCs (SONIC) Center, a 5-year multi-university center funded by DARPA and SRC under the STARnet program.