

A Low-Power Signal-Dependent Sampling Technique: Analysis, Implementation, and Applications

Ehsan Hadizadeh Hafshejani^{ID}, Mohammad Elmi, Nima TaheriNejad^{ID}, *Member, IEEE*,
Ali Fotowat-Ahmady, *Member, IEEE*, and Shahriar Mirabbasi^{ID}, *Member, IEEE*

Abstract—Sensors are among essential building blocks of any Cyber-Physical Systems (CPSs). Acquisition and processing of their sensory data contribute to the power consumption and computation load of the overall CPSs. For data acquisition, the conventional fixed frequency sampling in many such systems is sub-optimal since a sizable number of samples do not contain important information. In this work, we propose a Signal-Dependent Sampling (SDS) method and present its associated circuit implementation. Using the proposed SDS method, the number of retained samples is significantly reduced with little or negligible compromise in the quality of the (reconstructed) signal. The associated error and added noise are analyzed and their boundaries — which can be controlled by the user — are calculated. Our experiments show that the proposed system is able to improve power efficiency of the overall system in various applications. For example, for wireless Electrocardiography (ECG), Photoplethysmography (PPG), and Electroencephalogram (EEG) monitoring systems, the proposed approach can achieve a power saving of 81%, 76%, and 64% respectively. The proof-of-concept prototype system is implemented using TSMC 0.18 μm and has a foot-print and power consumption that compare favorably with those of the state-of-the-art implementations. The method can be used in a variety of applications including wireless sensor networks, mobile and wearable devices, as well as Internet of Things (IoT) nodes.

Index Terms—Signal-dependent sampling, non-uniform sampling, analog-to-digital conversion, low-power, wireless sensors.

I. INTRODUCTION

SENSORS have been pervasively used in the modern systems for various purposes. Acquiring and processing the (tremendous amount of) data produced by these sensors have attracted the attention of researchers [1]–[11]. In the past, a vast majority of conventional data acquisition approaches

have used uniform sampling, in which the signal is periodically sampled without taking into account the properties of the signal (other than its bandwidth); therefore, a noticeable amount of power is unnecessarily used for the sampling, processing, and transferring/saving the low-frequency contents or long periods of silence in the original signal [12]. As a consequence, the uniform sampling might not be the favorable optimum method in many applications [12]. For example, consider an Electrocardiography (ECG) biosignal where most parts of the signal have little to no activity and do not carry useful information. Thus, more efficient techniques such as Compressed Sensing (CS) [13], [14], digital compression [9], and nonuniform sampling methods [15] such as event-based sampling techniques [1], [2], [12] have been used to alleviate the cost of Nyquist-rate sampling.

Reconstructing the original data can be achieved by employing various methods through which the number of sampled data can play a significant role in reducing the power consumption and the quality of the recovered signal. In most digital compression techniques, first, the original data is uniformly sampled at the Nyquist-rate [9], then digital data compression techniques are used to reduce the volume of the amount of data required for a legible representation of the signal. The limitations of uniform sampling can be still costly in such methods. On the other hand, CS based sampling allows sampling at a rate that can be lower than the Nyquist-rate and is proportional to the informative content of the signal. The CS can be specifically useful for sparse signals [16]. There might be some trade-offs in CS based designs due to the limitations such as complexity [17] and sensitivity to noise [18].

In an event-driven non-uniform sampling, the signal is sampled only when a sizable change, *an event*, has occurred in the original signal. The aforementioned change can be, for example, in the amplitude or the phase level of the signal. This concept has been often used in level-crossing sampling [4], [6], [19]–[21] and continuous-time quantization techniques [12]. In these techniques, specific levels are often predetermined (or can be adjusted during run-time), and the data is sampled only when the signal crosses the aforementioned levels. Considering the amplitude or phase dependency of the approach, the reconstructed signal may not contain some of the information content of the original signal, especially if the separations among the levels are not fine enough. Adaptive

Manuscript received May 20, 2020; revised August 9, 2020; accepted August 20, 2020. Date of publication September 11, 2020; date of current version December 1, 2020. This work was supported by the Natural Sciences and Engineering Research of Canada (NSERC). This article was recommended by Associate Editor A. Worapishet. (*Corresponding author: Ehsan Hadizadeh Hafshejani.*)

Ehsan Hadizadeh Hafshejani, Mohammad Elmi, and Ali Fotowat-Ahmady are with the Sharif University of Technology, Tehran 11365-11155, Iran (e-mail: ehsan.hadizadeh89@gmail.com).

Nima TaheriNejad is with the Department of ICT, TU Wien, 1040 Wien, Austria.

Shahriar Mirabbasi is with the Electrical and Computer Engineering Department, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada.

Color versions of one or more of the figures in this article are available online at <https://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSI.2020.3021290

1549-8328 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See <https://www.ieee.org/publications/rights/index.html> for more information.

level crossing techniques have been used to make changes to the levels to improve the sampling efficiency of the overall system.

Other solutions for reconstructing a signal from a lower number of samples as compared to that of Nyquist-rate sampling have been proposed. For example, one can use the least-squares (LS) method [22] to fit a line to a set of consequent samples and approximate the resulting LS line with only two points. However, such solutions need complicated and power-hungry calculations. In this article, we present an alternative event-driven method for non-uniform sampling of data. In this approach, based on the previous signal samples, we predict the value of the signal at the current sampling point [23], [24]. An event has occurred when the error between the predicted value and the original signal level at the current sampling instant is larger than a predefined threshold. As will be described later, in the proposed technique, the rate of change in the signal determines the number of samples that are used for processing. Therefore, when there are no changes or slow changes in the signal no samples are retained or post-processed, which leads to a considerable saving in the number of retained samples as well as the overall power consumption of the system. Another important feature of the proposed method is the ease of adjusting its threshold which in turn controls the amount of reconstruction noise.

The proposed method is designed, implemented and successfully tested using a compact ultra-low-power mixed-signal system. In this system, the sampling process and decision-making about keeping the sample are done mainly in the analog domain without invoking the Analog-to-Digital Converter (ADC) to convert every sample. In this way, not only the average power consumption of the ADC is reduced by the Compression Factor (CF), but also the subsequent blocks in the system which are usually power hungry (such as processor and/or RF front-end) can stay in standby mode when there is no converted sample available which would result in a lower overall power consumption of the system. The proposed sampling circuit is used in a variety of applications such as ECG, Photoplethysmography (PPG), and Electroencephalogram (EEG). For example, using the proposed sampling circuit, the number of samples (and the consequent power consumption) of an ECG sensor are reduced by 86% and 81% as compared to those of Nyquist sampling, while maintaining around a reasonable Post-Reconstruction Signal-to-Noise and Distortion Ratio (PR-SNDR) of 30 dB. Note that the proposed technique can be used in conjunction with other uniform sampling ADCs to reduce the power of the ADC as well as the power of the subsequent data processing/transmission units.

The rest of the paper is organized as follows: in Section II we introduce the proposed method and its associated noise analysis. Next, we present the details of the Integrated Circuit (IC) implementation in Section III. This is followed by Section IV in which we present the simulation and measurement results of the fabricated chip. In Section V, we present the performance of the proposed method for different applications. We then compare the performance of the proposed circuit with

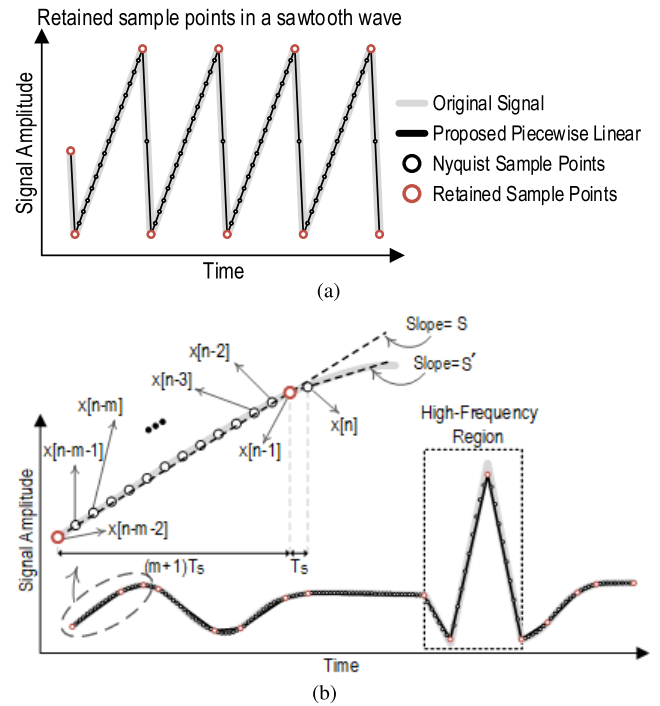


Fig. 1. An exemplary signal to visualize the proposed sampling method.

that of similar works in Section V-D and provide concluding remarks in Section VI.

II. THEORY

A. The Proposed Method

Let us consider the triangular (saw-tooth) signal in Fig. 1.a. Because of the relatively fast rate of change of signal, its spectrum contains high-frequency components. Thus, the Nyquist sampling rate required to sample the signal is rather high (which is higher than the fundamental frequency of the triangular wave). Assume that instead of retaining all the sampled values, we just keep the depicted points (the red points in Fig. 1.a) and the time intervals between the points. In this way, using linear interpolation we can recover all other samples in between. In other words, such an approach approximates the signal with its piece-wise linear representation (i.e., with a number of consequent linear segments). In the proposed approach, the input signal is sampled at the Nyquist rate or more, and the consecutive samples which are lying on a given linear segment can be approximated based on the two endpoint samples of the segment and the corresponding time difference between the two endpoints (Fig. 1.b). The decision on which samples (endpoints of each line segment) to keep is made based on the approximation of the second derivative of the signal waveform and does not depend on the signal spectrum but on the shape of the signal. For example, for a periodic saw-tooth signal, independent of the signal frequency, only the minimum and maximum points along with their time difference are kept. An example of a more complex signal and the retained samples using the proposed method are shown in Fig. 1.b. As we can see in this figure, depending on the shape of the signal a larger or smaller number of samples between retained points (shown in red) can be dropped without significant information loss.

In what follows we describe the algorithm that is used to decide whether or not to retain a sample point. As mentioned earlier, in the proposed technique, the signal is represented by a piece-wise linear waveform. For each line segment, the endpoint samples as well as the information regarding the time interval between the two endpoints are stored. Let us assume that a Nyquist or higher rate sampling clock, denoted by CLK with a period of T_s is used to sample the signal. To decide whether or not to keep a sample, the proposed algorithm uses the following three samples. One is the current input sample, i.e., $x[n]$. (Note that $x[n]$ refers to the signal sampled at nT_s , i.e., $x(nT_s)$ and for the purpose of brevity, we follow the common practice and drop the T_s from the notations.) The second is the previous sample immediately taken before the current sample, that is, $x[n-1]$. The third important sample is the last retained sample, namely, $x[n-m-2]$, where m is the number of discarded samples in the corresponding line segment. At any given time, n , the system uses $x[n-m-2]$, $x[n-1]$ and $x[n]$ to determine whether to retain $x[n-1]$. This decision is made based on

$$\left| \frac{x[n-1] - x[n-m-2]}{(m+1) \times T_s} - \frac{x[n] - x[n-1]}{T_s} \right| \leq \varepsilon \quad (1)$$

where the first term on the left-hand side of the inequality represents the relative slope of the line segment between $x[n-1]$ and $x[n-m-2]$ (S in Fig. 1.b) and the second term indicates the slope between $x[n]$ and $x[n-1]$ (S' in Fig. 1.b). Also, ε is the error tolerance which we refer to as the fidelity bound. If the two slopes are close ($|S - S'| \leq \varepsilon$), $x[n-1]$ can be discarded. Otherwise, $x[n-1]$ will be retained and becomes the starting point of the next segment. Discarding the samples results in the reduction in the number of stored data needed to reconstruct the sampled signal. To reconstruct the signal, the discarded samples are reconstructed using linear interpolation. Such linear interpolation could result in error between the reconstructed sample and the original discarded sample. The analysis provided in the next section provides a measure for the maximum reconstruction error.

B. Added Error Analysis

Similar to the approach used for the calculation of the quantization noise in ADCs, we can model the error between the reconstructed sample and the original discarded sample as an added noise. To this end, we calculate the worst case error for each sample and model these errors as a Root-Mean-Square (RMS) noise that is added to the reconstructed signal.

Assume a signal which is sampled with a sampling clock, CLK, and the samples are such that the left-hand side of (1) is equal to ε (i.e., worst-case scenario) as shown in Fig. 2. In this case, the sampling unit after saving the initial sample point (x_0), to make a decision about the next sample point (x_1), will compare the slopes of lines S_1 and S'_1 . Here, we assume the difference between these two slopes is equal to ε to present the worst-case scenario. In this case, the sampler will drop x_1 . For deciding about the next sample point (x_2), the slopes of lines S_2 and S'_2 will be compared and since we have assumed that (1) is met, x_2 will be dropped. This will continue and all the samples will be dropped one after another.

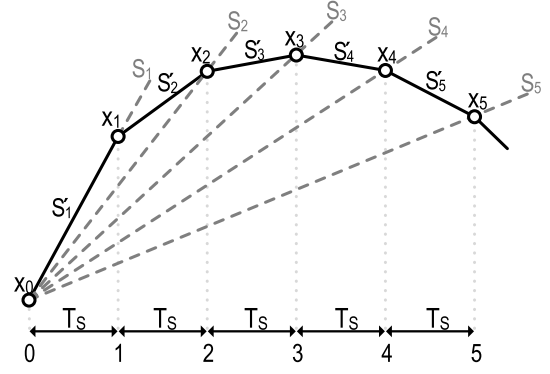


Fig. 2. Representative samples of the input signal and associated piece-wise linear segment slopes for worst-case scenario.

As we can see in Fig. 2, the deviation from the original signal and the estimated line which connects the first and last sample, becomes larger and larger as we increase the number of dropped samples. Although the probability of having such a signal for which all the samples meet the condition in Fig. 2 is very small, however, we can control the reconstruction error in such a worst-case scenario by limiting the maximum number of consecutive samples which can be dropped. We define N as the limit on the maximum number of samples that could be ignored ($m < N$). Given ε and N , one can make sure that the reconstruction error is within an acceptable range. To find the difference between the values of the dropped samples, (x_i), and the estimated values of the reconstructed samples (\hat{x}_i), we first write the expression of the values of the original samples.

$$\begin{aligned} x_1 &= x_0 + S_1 \times T_s, \\ x_2 &= x_1 + S'_2 \times T_s, \\ &\vdots \\ x_i &= x_{i-1} + S'_i \times T_s \end{aligned} \quad (2)$$

where T_s is period of the sampling clock ($T_s = \frac{1}{f_s}$), S'_i is the slope of the line between sample x_{i-1} and sample x_i that based on the assumption of worst-case scenario, we will then have:

$$S'_{i+1} = S_i - \varepsilon. \quad (3)$$

We know that S_i is the slope of line between sample x_0 and sample x_i (slope of the estimated line in each step) and can be written as

$$S_i = \frac{x_i - x_0}{i \times T_s}. \quad (4)$$

Combining (2) to (4), we have:

$$\begin{aligned} S_i &= \frac{(S_{i-1} \times (i-1) \times T_s + (S_{i-1} - \varepsilon) \times T_s + x_0) - x_0}{i \times T_s} \\ &\Rightarrow S_i = S_{i-1} - \frac{\varepsilon}{i} \\ &\Rightarrow S_i = S_1 - \sum_{j=2}^i \left(\frac{\varepsilon}{j} \right). \end{aligned} \quad (5)$$

Then we can rewrite (4) as:

$$x_i = \left(S_1 - \sum_{j=2}^i \left(\frac{\varepsilon}{j} \right) \right) \times i \times T_s + x_0 \quad (6)$$

(6) provides the values of original samples of the signal in the above-mentioned worst-case scenario. Now, assuming the linear interpolation reconstruction (other reconstruction methods such as splines [25] can also be used for minimizing the reconstruction error, without the loss of generality) One can calculate the value of reconstructed sample points, (\hat{x}_i) , as follows:

$$\begin{aligned}\hat{x}_i &= S_N \times i \times T_s + x_0 \\ \Rightarrow \hat{x}_i &= \left(S_1 - \sum_{j=2}^N \left(\frac{\varepsilon}{j} \right) \right) \times i \times T_s + x_0\end{aligned}\quad (7)$$

Now, we can calculate the added error to the signal by subtracting (7) from (6):

$$|e_i| = |x_i - \hat{x}_i| = \left(\sum_{j=i+1}^N \left(\frac{1}{j} \right) \right) \times i \times T_s \times \varepsilon \quad (8)$$

where e_i 's are the added errors to each sample after reconstruction. As it can be seen in (8), the added error to the signal has a direct relationship with ε . To illustrate the relation between e_i and N for the signal shown in Fig. 2, we calculate the added error to each sample in the worst-case scenario using *Matlab* from (8) for $\varepsilon \times T_s = 1 \text{ mV}$ and various values of N .

Based on the expression of the error derived in (8), one can calculate the RMS of the added noise as:

$$\overline{V_n^2} = \overline{e^2[i]} = \frac{1}{N \times T_s} \sum_{i=0}^N e^2[i] \quad (9)$$

$$V_{n,rms} = \sqrt{\frac{1}{N \times T_s} \sum_{i=0}^N e^2[i]} \quad (10)$$

where $V_{n,rms}$ is the maximum RMS added noise to the signal due to the proposed Signal-Dependent Sampling (SDS). To find an upper bound for Equation 10, we assume that T_s is small and re-write the equation in the following integral form:

$$V_{n,rms} = \sqrt{\frac{1}{T} \int_0^T \left(i \times \varepsilon \times \int_i^T \frac{dj}{j} \right)^2 di} \quad (11)$$

where T is the maximum time during which no samples are retained, i.e., $T = N \times T_s$. This gives us

$$V_{n,rms} = \sqrt{\frac{\varepsilon^2}{T} \int_0^T \left(i \times (\text{Log}(T) - \text{Log}(i)) \right)^2 di} \quad (12)$$

and can be rewritten as:

$$V_{n,rms} = \sqrt{\frac{\varepsilon^2}{27T} \left[i^3(2 + 9 \text{Log}(i)^2 + 6\text{Log}(T)) + \dots \right.} \\ \left. \frac{9 \text{Log}(T)^2 - 6 \text{Log}(i)(1 + 3 \text{Log}(T))}{9} \right]_0^T}$$

which can be further simplified to

$$V_{n,rms} = \sqrt{\frac{2\varepsilon^2}{27T} T^3} = 0.27 \times T_s \times \varepsilon \times N \quad (13)$$

Equation 13 gives us the upper bound for the error in worst-case scenario but for real signals the error is much

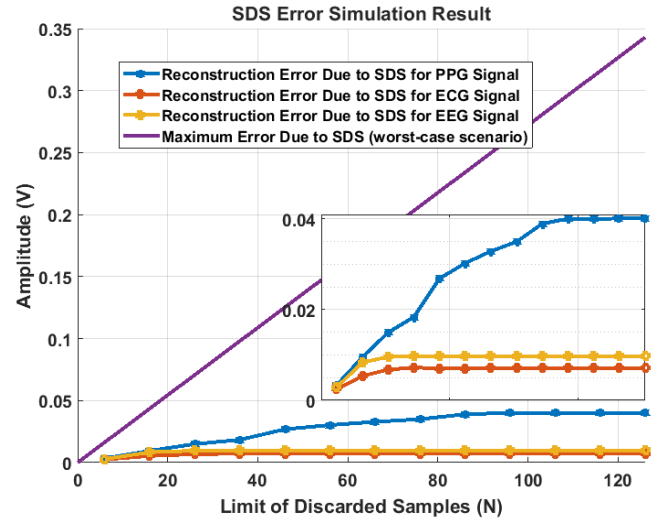


Fig. 3. Reconstruction added RMS noise to the various signals.

lower than the value predicted by this equation. Fig. 3 provides examples of real signals and their added error compared with this worst-case scenario. As it can be seen from the figure, based on the nature of the signals the reconstruction error is saturated at some point and does not grow with N .

C. Decision-Making Process

In the proposed method for an input signal sampled at the Nyquist or a higher rate, if some consecutive samples fall on the same line, we only store the first and last sample of the line and discard all the other intermediate samples. To have a (near) real-time sampling, ideally, we need to make a decision to whether keep or discard each new sample immediately after the sample is taken. Hence, we need a decision-making rule which does not require complicated calculations. Moreover, in a causal system this decision should be made using only current and past samples. Hence, we propose the following decision-making scheme.

In Fig. 1, we assume that $x[n-m-2]$ is the “last retained sample” and m samples (d_1 to d_m) are being discarded. Thus, we need to decide whether to keep $x[n-1]$ and to make this decision we need the information about the slope of the line segment between $x[n-1]$ and $x[n]$. As shown in Fig. 4, by calculating the slope of the line between $x[n]$ and $x[n-1]$, namely, S' , and comparing it with S (which is the slope of the line between $x[n-1]$ and $x[n-m-2]$), we can find out whether the position of $x[n]$ is within an acceptable tolerance. That is, if S' is equal to S or is within an acceptable error bound, ε , from it, then we will discard $x[n-1]$. Otherwise, $x[n]$ is not on the estimated line and therefore, $x[n-1]$ should be stored (retained) as the last point of the previous line segment. Thus the value of the last retained sample will be updated from $x[n-m-2]$ to $x[n-1]$ and the counter for number of discarded samples will be reset.

III. CIRCUIT DESIGN

Given the overall goal of the proposed technique is to minimize the number of stored samples and reduce the power of whatever ADC or signal conversion that is to be used,

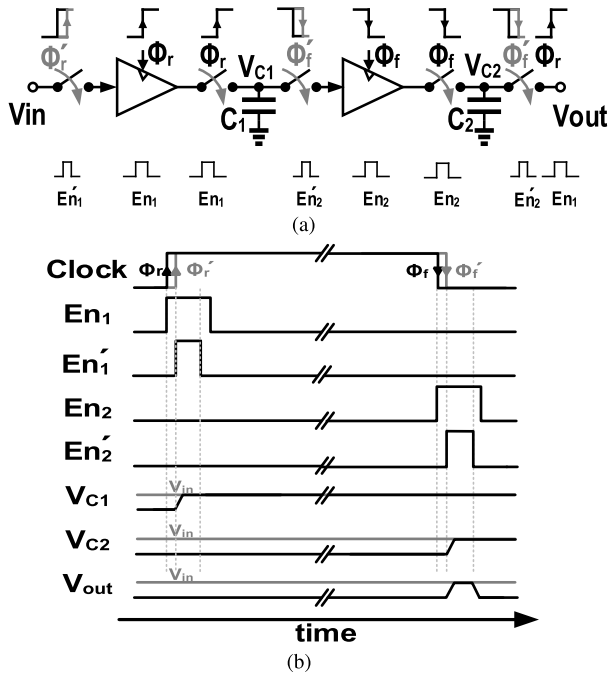


Fig. 7. Timing diagram of SH blocks. Enable signals with a small offset prevent charge sharing and unintended propagation of stored charge.

by En_1' (starting at Φ_r') and En_2' (starting at Φ_f') signals. In this way, when the buffer is turning on or off, the charges on the storage capacitor of the previous stage will see the high impedance of the off switch. Thus, minimal or ideally no charge injection will happen.

Using larger size switch transistors could result in a smaller “ON” resistance for the switches which reduces settling time of the sampled voltage. On the other hand, larger size switches result in larger parasitic elements which introduce their own drawbacks such as charge injection and clock feed-through. In this design, the leakage of “OFF” switches has the most dominant effect, especially in lower clock frequencies when the sampled voltage on the storage capacitor should be held constant for a long time. To minimize the adverse effect of the leakage, the storage capacitance should be chosen as large as possible. Using larger storage capacitors, at the cost of larger area chip, would also result in sufficiently low $\frac{kT}{C}$ noise, however, it also results in a larger RC time constant for charging the storage capacitors. This larger charge/discharge time could in turn result in additional side effects such as the capacitors not being fully charged during the enable phase of the switches. This can be ameliorated by extending the width of enable pulse or increasing the bias current of the buffers, at the cost of higher power consumption. Therefore, there is a trade-off among power, size of switches, size of storage capacitors, chip area, and the accuracy. If these parameters are not chosen properly, the power budget is affected and the allowed maximum number of discarded samples is reduced.

Stored values of the first SH and the second SH are updated at each clock cycle. However, the sampling clock of the third SH is the “ADC Trigger” signal and thus will only get updated when “ADC Trigger” is activated. Note that because of a longer period of holding phase on the third SH, its stored value may drop due to the leakage currents. Therefore, the storage

capacitors of the SH₃ must be larger than those of the other two SH stages. Furthermore, the buffers should be low-offset so as to minimize the adverse effects of their offset on the accuracy of the ADC. In other words, the added error due to the offset of the buffers should be lower than the Least-Significant Bit (LSB) of the ADC. In this design, the buffers are overdesigned to have an accuracy of just over 15 bits so that they do not adversely affect the accuracy of the overall system. The simulated input dynamic range of the overall SDS when operating with a 1 kHz clock and taking into account the adverse effects of leakage of the switches, offset of the buffers, charge injection and clock feedthrough and other parasitic effects is 78.27 dB. In this case, the full swing of the input signal is 1.8 V.

B. Subtractor Circuit

Differentiator blocks (subtractors) are used in the proposed system to calculate the difference between the voltages stored on SHs. Considering that the voltages of SHs can be within the full-scale range of the ADC, the input Common-Mode Range (CMR) of the subtractors should also cover the full-scale range of the ADC. Subtractors also should have a high Common-Mode Rejection Ratio (CMRR). Note that the common-mode values of the two inputs of the subtractor could be high, however, the difference the input signals is usually a small value, a low CMRR will affect the precision of the subtractor. To address this issue, we have used differential G_m structures for subtractors. Thus, we can set a high gain for the subtractors without undergoing saturation problems at the output. A simplified circuit diagram of the G_m block is shown in [26, Chapter 13, Page 460].

The output current can be written as:

$$i_{out} = g_m(V_{i+} - V_{i-}) \quad (14)$$

where g_m is the differential transconductance of the G_m block. The G_m blocks set the CMRR of the overall system and are designed to provide the overall CMRR of 86.05 dB.

In the proposed system, to calculate the slopes of the piece-wise linear segments, we do need to divide the voltage differences by the associate time difference between the samples. The time difference between V_1 and V_2 (which are consecutive) is equal to one clock period (T_s). And the time difference between V_2 and V_3 is equal to $(m+1) \times T_s$ (note that V_1 , V_2 , and V_3 are $x[n]$, $x[n-1]$, and $x[n-m-2]$, respectively). To measure the time difference between samples V_2 and V_3 we need to count the number of the discarded samples. In the next section, we will explain the counting and dividing process.

C. Counter and Divider

To keep track of the number of samples that should be discarded we use a binary counter which counts up at the rising edge of the sampling clock. The value of this counter increases when a new sample, i.e., $x[n]$ is taken and the systems decides to drop the previous sample, i.e., $x[n-1]$. The counter output is also used for calculating the derivative (slope) of the signal and the value will be stored when the proposed ADC decides to

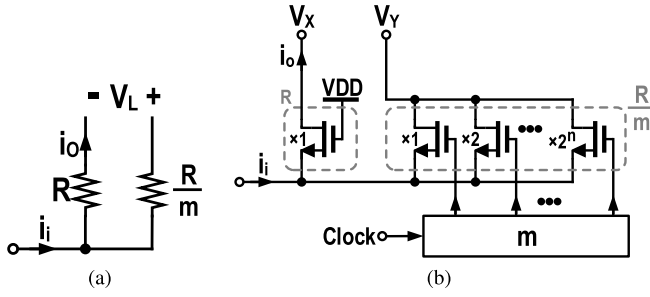


Fig. 8. The binary weighted resistors used in this design. a) Ideal current divider. b) Implementation of the current divider using triode region MOS transistors.

retain a sample. The stored values of the counter would also be used in the signal reconstruction. The counter will “RESET” when the previous sample, that is, $x[n-1]$ is retained.

Using the subtractors, G_m blocks, the values of $g_m(V_1 - V_2)$ and $g_m(V_3 - V_2)$ are calculated. To find the slope of the line between samples V_2 and V_3 , we need to divide the output of G_{m2} by the number of clock periods between these samples, which is provided by the counter. With a high precision divider circuit, the output current of the second G_m block can be divided by $m+1$, however, the output of the G_m block is an analog current and the output of the counter is a digital signal. Therefore, the divider ideally should be able to divide an analog current by an integer number. For this division, we use the concept of current division as shown in Fig. 8(a). The input current i_i is divided between the two resistors. If the voltage V_L is zero, the two resistors will be in parallel and the output current, i_o , would be:

$$i_o = \frac{\frac{R}{m} \times i_i}{\frac{R}{m} + R} = \frac{i_i}{m+1}. \quad (15)$$

To implement resistors R and R/m , and to further save area, we use transistors that are operating in their triode region (refer to Fig. 8(b)). To ensure that these two resistors are (effectively) in parallel, they should have the same voltage across them. To implement R/m we use binary-weighted switches which are driven by the counter output. These binary-weighted transistors operate in the triode region (shown in Fig. 8(b)) and their equivalent resistor is R/m . A matched transistor (with a size equal to the smallest switch of the binary-weighted array) is used to implement R .

Thus, in Fig. 8(b), i_o is proportional to the slope of the line segment between V_2 and V_3 . To implement (1), we need to find a difference between two slopes. This is implemented using the circuit shown in Fig. 9. In this circuit, because of the negative feedback of the Operational Amplifier (OpAmp), the negative input, V_X is at the same voltage as that of the positive input, i.e., $V_X = V_Y = V_{ref}$. Therefore, all transistors taking part in the current division are effectively in parallel (i.e., have the same voltage across them). Additionally, we have a current summation (between i_{o1} and i_{o2}) in node X. The low input impedance of the Transimpedance Amplifier (TIA), R_{in} , will ameliorate the adverse effects of the parasitics at node X and $i_{o1} + i_{o2}$ will flow to the R_f . V_{fed} is the voltage which can be

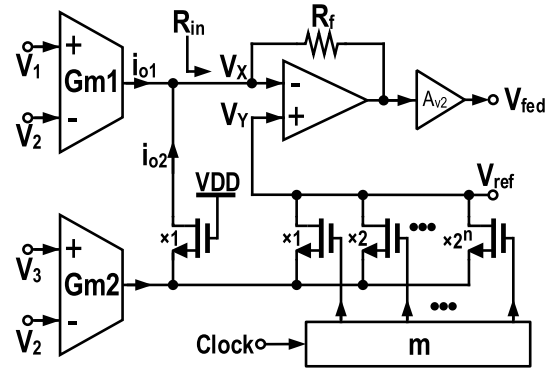


Fig. 9. The circuit for implementing left-hand-side of expression (1).

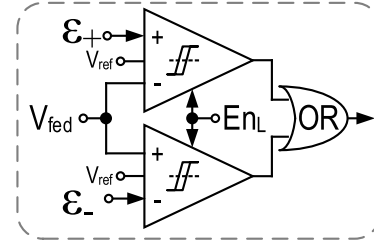


Fig. 10. Block diagram of the implemented two-level comparator.

used for comparison purposes and can be written as:

$$V_{fed} = V_{ref} - \left[(V_1 - V_2) - \frac{(V_2 - V_3)}{m+1} \right] \times g \times R_f \times A_{v2} \quad (16)$$

The output of this circuit, namely, V_{fed} , is thus proportional to the left-hand-side of (1).

We note that transistor mismatch and leakage of OFF switches can affect the accuracy of the divider. Hence, the size of the transistor and the layout of the divider should be designed attentively to minimize these effects.

D. Comparator

The circuit in Fig. 9 provides a voltage (V_{fed}) that is proportional to the left-hand-side of (1). Fig. 10 shows a simple circuit that implements the “absolute value” function and the comparison needed to evaluate (1). Let us define:

$$\varepsilon_v = k \times \varepsilon \times T_s \quad (17)$$

where $k = g \times R_f \times A_{v2}$ is the gain of the circuit shown in Fig. 9. To compare $|V_{fed}|$ with ε_v we have used a combination of two comparators to implement a two-level comparison as shown in Fig. 10. In addition to supporting both positive and negative comparison levels, the positive and negative comparison levels ($\varepsilon_+ = \varepsilon_v$ and $\varepsilon_- = -\varepsilon_v$) can have different absolute values for more flexibility, that is, the positive and negative limits can have different magnitudes.

Finally, as shown in Fig. 5, the output voltage of the two-level comparator of Fig. 10 will be logically “AND”ed with the clock to produce the *ADC Trigger* signal. At the rising edge of the clock signal, *ADC Trigger* indicates whether the associated sample (sample on the second SH) must be retained (when *ADC Trigger* is “1”) or discarded (when *ADC Trigger* is “0”). If the sample is supposed to be retained, it will be

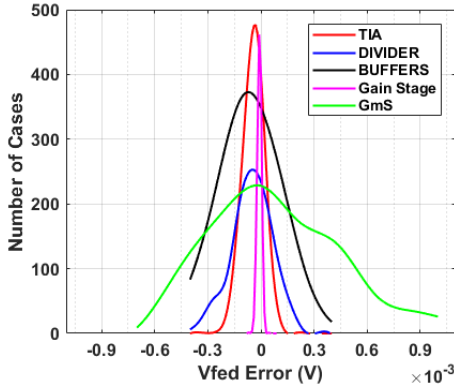


Fig. 11. Error imposed on V_{fed} by each individual sub-circuit due to PVT variations.

converted to digital using the ADC and the corresponding counter value will be stored. As shown in Fig. 5, when a sample is retained, the circuit will “RESET” the counter and store the retained sample into the third SH.

E. The Effects of PVT variations and Noise

As in any practical circuit, in the proposed design, Process-Voltage-Temperature (PVT) variations (and the associated offset) and noise of the circuit blocks can have adverse consequences. For example, in our system they may affect the decision making process and cause deviations from the expected results. In this subsection, we review these effects on the overall performance of the system through Monte Carlo simulations. Fig. 11 shows the error in V_{fed} that is produced by each individual block, namely, the sample and hold buffers, G_m blocks, TIA, the gain stage, and the divider transistors. These errors are due to PVT variations and are calculated using Monte Carlo simulations of 1000 random cases. As it can be seen from the figure, the maximum error is produced by the G_M blocks and is an order of magnitude lower than the typical values of V_{fed} . Fig. 12 shows the performance of the whole SDS system in the presence of PVT through Monte Carlo simulations of 5000 random cases. More specifically, Fig. 12(a) and Fig. 12(b) show the effects of PVT on PR-SNDR and CF, respectively. The input signal in these simulations is an ECG signal. The brown bars show the performance for ideal case and the blue bars show the performances in the presence of PVT. As it can be seen in Fig. 12, the circuit is robust against PVT variations, that is, PR-SNDR and CF are relatively constant and in the majority of cases produce the nominal value. Another consideration is the effect of Gaussian noise on the performance of the circuit. This noise can be due to the noise of the input signal or the circuit. Noise can cause false triggers that can affect the CF and PR-SNDR of the sampled signal. Regarding Equation 1, we can define a condition that guarantees minimum false triggers:

$$\frac{2 \times V_{p-p}}{T_s} \leq \varepsilon \quad (18)$$

where V_{p-p} is peak-to-peak voltage of the noise. To show the robustness of the design, we have added random Gaussian noise to the system and have swept the RMS voltage of the noise. Fig. 13 shows the performance of the SDS, namely,

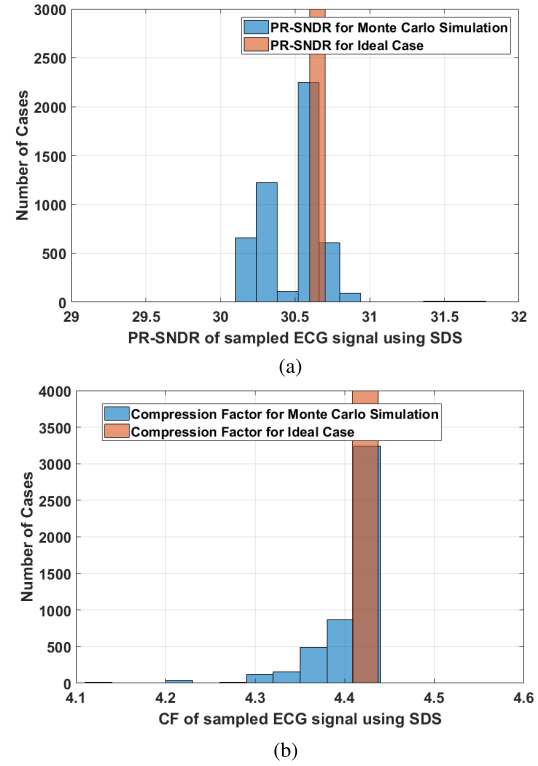


Fig. 12. Effect of PVT variations on the performance of the SDS. Input signal is ECG, $\varepsilon \times T_s = 20mv$, and $N = 9$. a) Effect on PR-SNDR of ECG signal for 5000 random cases. b) Effect on CF of ECG signal for 5000 random cases.

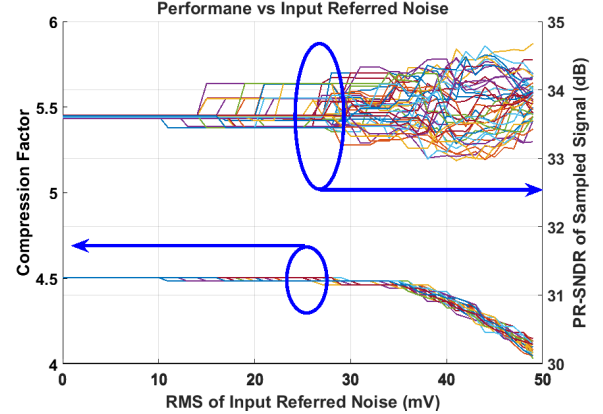


Fig. 13. Effect of Gaussian noise on the performance of SDS when the input signal is ECG. This test is done for 50 different simulations, $\varepsilon \times T_s = 100mv$ and $N = 8$.

PR-SNDR and CF as a function of RMS of the added noise. These plots are the results of 50 different simulations where $N = 8$ and $\varepsilon \times T_s = 100 mV$. By a conservative assumption of $V_{p-p} = 4V_{n,rms}$ for Gaussian noise to cover more than 95% of situations, we expect that we have no (less than 5%) false trigger up to $V_{n,rms} = 12.5 mV$ and by assuming $V_{p-p} = 2V_{n,rms}$, we expect that CF and PR-SNDR be almost constant up to $V_{n,rms} = 25 mV$. As it can be seen from the figure, the circuit is robust to noise (no changes in CF and PR-SNDR), when RMS of the added noise is less than 10 mV and similarly, CF is almost constant when the RMS of the added noise is less than 25 mV. Even when the noise increases beyond that, the variations in PR-SNDR and CF are not drastic.

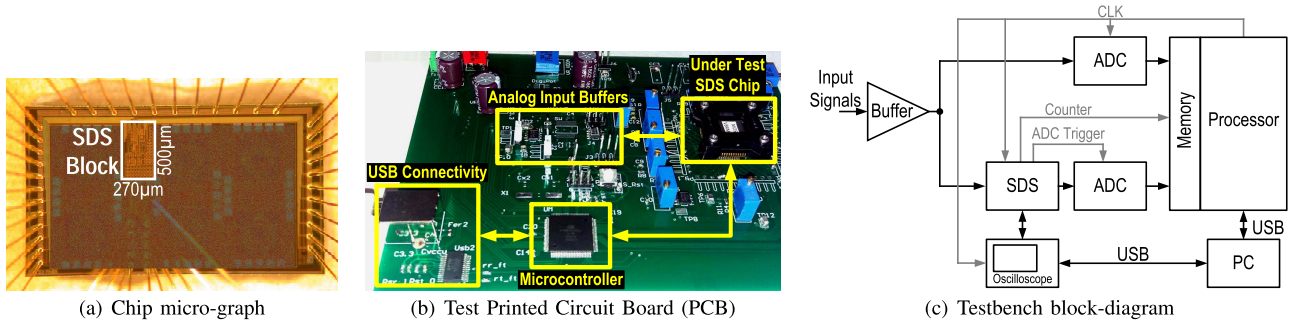


Fig. 14. Test environment of the proposed SDS technique.

TABLE I
CHARACTERISTICS OF THE USED ADC

Parameters	Value	Units
Type	SAR ADC	
Resolution	12	Bit
INL	± 3	LSB
DNL	> -1	LSB
Conversion Time	2.1	μS
Energy Consumption	680	fJ/Bit

IV. EXPERIMENTAL RESULTS

A. Implementation and Measurements

The proposed SDS system (Fig. 5) is implemented in a $0.18\text{-}\mu\text{m}$ CMOS technology. The fabricated system occupies 0.135mm^2 of silicon area ($500\mu\text{m} \times 270\mu\text{m}$). Fig. 14(a) shows the chip micrograph.

System evaluations are performed by sampling the input signal simultaneously by both a Nyquist-rate ADC and the proposed SDS system. The test-bench setup is shown in Fig. 14. In the two paths of the test bench, two identical ADCs are used so that the comparisons of CF and power consumption are fair. The characteristics of the off-chip ADC that is used in this work are summarized in Table I. In each test, a clock frequency (CLK) suitable for the input signal is used. The first ADC outputs the uniformly sampled data without applying any compression technique. At the same time, the second synchronized ADC uses the proposed non-uniform sampling approach, i.e., SDS to sample the input signal. Some critical signals such as output voltages of the SHs (V_1 , V_2 , and V_3), enable signals (EnH , EnL ,...), and input voltage of the comparator (V_{fed}) are brought out of the chip so that they can be observed using a digital oscilloscope. The oscilloscope trigger signal and ADCs sampling clocks are all synchronized with the CLK signal which is generated by the microcontroller. In this proof-of-concept prototype, the reference voltage of the comparator (V_{ref}) and the comparison levels (ε_+ and ε_-) are generated off-chip and are inputted to the test chip. *Matlab* software has been used to analyze the collected data.

As mentioned earlier, active parts of the circuit are turned on at the corresponding clock edges. The enable signals are generated on the test chip and the duration of these pulses can be controlled using the associated control inputs of the chip. Based on the maximum settling time in all parts of the circuit, which is around $5\mu\text{s}$, the duration of enable signals in normal operation is set to $10\mu\text{s}$. This would guarantee that all parts of the circuit will work properly. Furthermore, due to the low duty cycle of all the building blocks of the proposed circuit,

the overall power consumption is kept low. For example, for $\text{CLK}=1\text{ kHz}$, the duty cycle of the circuit is 1%. However, to show more details of the operations, in the performance measurements reported in the following subsection, we set the duty cycle to a time comparable with the clock period (5% of the CLK period). This helps us demonstrate how the circuit is working and what is happening in the important parts of the circuit during several clock periods. The clock frequency in these tests (reported in the next subsection) is set to 100 Hz .

B. Performance Summary

To evaluate the performance of various parts of the circuit, we applied an arbitrary signal to the input of the SDS. The output voltages of SHs are shown in Fig. 15(a). Each SH samples its input at the rising edge of its clock and their outputs would be updated at the falling edges of the clock. V_1 , V_2 , and V_3 are outputs of SH₁, SH₂, and SH₃ respectively. V_{S1} is a signal that is used to show the sampled version of the input signal (V_{in}). As it can be seen from the figure, the output voltages of SHs are valid for a short period of time (for the duration of enable signals). As expected, in contrast to V_3 , V_1 and V_2 are updated at each clock cycle.

Two other important building blocks of the circuit are the counter and the divider. Based on (16), by keeping $V_1 = V_2$ and $V_3 = V_2 + \Delta V$, V_{fed} would be equal to $g \times R_f \times A_{v2} \times \Delta V / (m + 1)$. So we can see the performance of the counter and divider as number of skipped samples (m) increases at each clock period. As shown in Fig. 15(b), by increasing the counter output from 0 to 5, V_{fed} (relative to V_{ref}) decreases from $0.2V$ to $0.033V$.

The final stage of the system is the decision making block. As explained before, to retain a digitized sample, one of the following two criteria should be met. Either the current sample deviates too much from the piece-wise linear approximation of the recent sequence of samples (i.e., $V_{fed} - V_{ref} > \varepsilon_+$ or $V_{fed} - V_{ref} < \varepsilon_-$) or the number of samples that are being dropped reaches N (the maximum allowed by user). Fig. 16 shows circuit performance for a time duration when $N = 8$ and $\varepsilon_+ = -\varepsilon_- = 200\text{ mV}$. There are three pulses generated on *ADC Trigger* output, as can be seen in the figure, the first pulse is generated because the counter has reached N . In the second case, $V_{fed} - V_{ref}$ has reached ε_- . Bear in mind that the calculation of V_{fed} occurs at the falling edge of the previous clock cycle and associated pulse generation is synchronized with the rising edge of the clock. The third case happens due to the first criteria mentioned

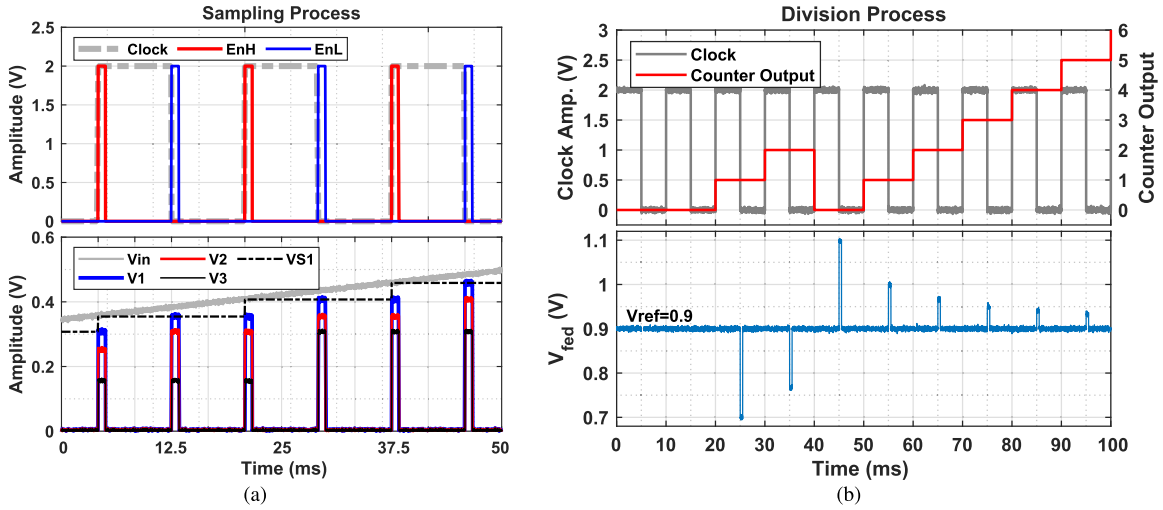


Fig. 15. An example of (a) sampling process (b) division process (measurement).

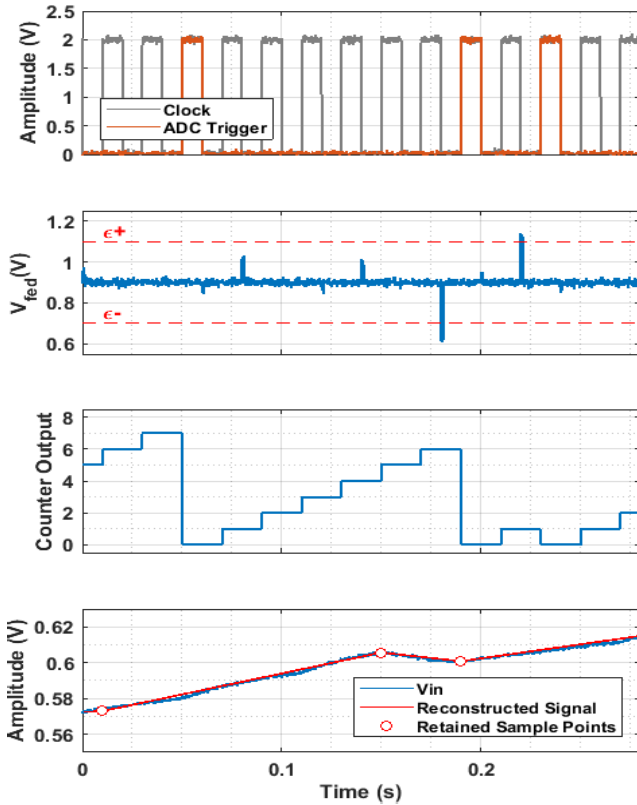


Fig. 16. Sampling triggered by maximum tolerated error or maximum number of skipped samples.

above, that is, $V_{fed} - V_{ref}$ has reached ϵ_+ . A point to consider here is the three delay components for processing the retained sample. The first delay component is due to the SH since the SH samples the input signal at the rising clock edge and places the sample at its output on the falling edge. Therefore, there is a “half clock period” processing delay here. The second delay component is due to the fact that the system is always making a decision regarding the previous sample i.e., $x[n-1]$. This corresponds to a “one clock period” delay. The third delay component is due to the fact that the system makes its decisions at the falling clock edge and waits for the rising clock edge to generate *ADC Trigger* pulse which is

synchronized with the rising edge of the clock. This accounts for another “half clock period” delay. Hence, as it can be seen in Fig. 16, when the system generates an *ADC Trigger* pulse, the ADC converts the value of the input signal that corresponds to “two clock periods” before the current clock. This sample is conveniently held at the output of SH₂ at the current clock. Thus, the delay between the input of the Nyquist ADC and the input of the ADC driven by the SDS circuit is exactly two clock cycles which can be compensated for in digital post-processing.

C. Power Consumption

Fig. 17(a) illustrates the measured total supply current drawn at different frequencies along with its predicted and simulated amount. Since the enable signals are generated at clock edges and their pulse-widths are constant (i.e., $10\mu s$ in this design), as clock frequency increases, the sleep time of the system decreases. Hence, we expect an increase in the drawn current that is proportional to the clock frequency (dotted line in Fig. 17(a)). The predicted supply current corresponds to the simulation and measurement lines except at very low frequencies. This deviation can be attributed to the static leakage power which becomes dominant at low frequencies. Fig. 17(b) presents the detailed information of the current drawn by each block at 1 kHz clock for ECG input signal. In this case, the CF is $\tilde{0}.1$, so the ADC is working at around 160 S/s. The reader may be wondering whether by implementing our proposed SDS with a fully digital circuit after a Nyquist-rate ADC. In the former approach, the proposed circuit can be implemented in a fully digital fashion. The digital and analog implementations have their own advantages and disadvantages. In terms of area and scalability, the digital implementation is advantageous, however, in terms of power consumption the analog implementation is more efficient. The power consumption of analog and digital implementations in the same technology are shown in Fig. 17(a). As can be seen from the figure, over all frequencies the power consumption of the analog implementation is lower than that of the digital implementation. Please note that Fig. 17(a) only shows the power consumption of the SDS unit. If we

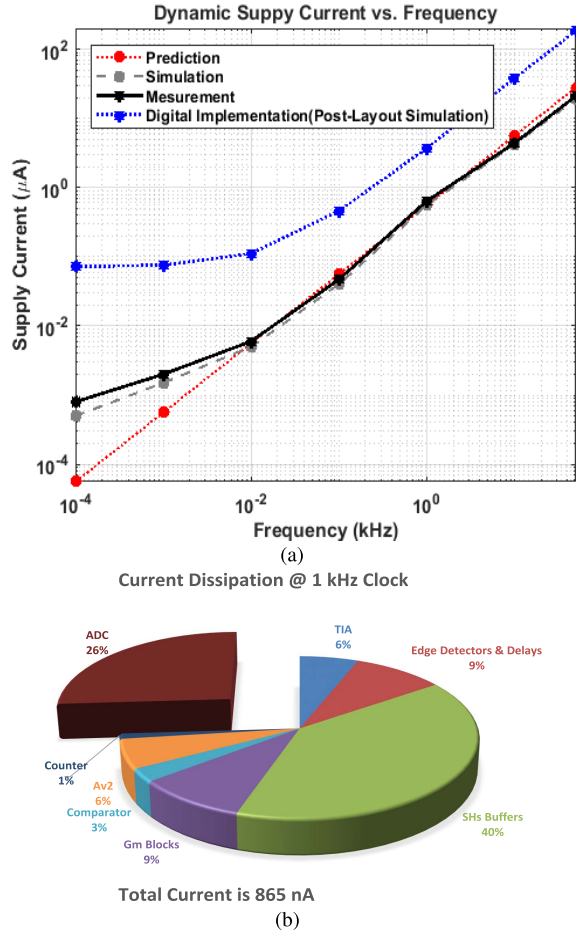


Fig. 17. (a) Current dissipation of the system based on the frequency of sampling, (b) current dissipation of various units and their role in the overall current dissipation of the system. Considering CF, the average conversion rate of the ADC is around 160 S/s.

include the power of the ADC, the power advantage of the analog implementation would be even more pronounced as in the digital implementation the ADC needs to operate at the Nyquist-rate. However, in more advanced technologies, one may be able to lower the power consumption of the ADC as well as the digital implementation of SDS, and thus it may worth implementing the system in the digital domain. However, the important point is that in either analog or digital implementation, using SDS would result in power saving in the overall system, especially in wireless systems where the samples are transferred wirelessly.

V. EXAMPLE APPLICATIONS

After testing all parts of the circuit in detail, we performed extended tests on the overall system using real signals. The performance of the system and the effect of N and ε on various signals, such as sine wave, saw-tooth, ECG, EEG, and PPG have been tested. For the purpose of brevity, here we report the effects of the parameters on digitizing a PPG signal. The results for other signals are shown for selected combination of parameters. The performance of the system is also tested for sine wave and saw-tooth signals in both high SNR and low SNR situations. For the high SNR sine wave (input SNR = 49.6), CF = 8.3 and PR-SNDR = 27.7 dB are achieved, and

for the high SNR saw-tooth (input SNR = 50.1), the system achieves CF = 13.7 and PR-SNDR = 39.5 dB. As observed in these two examples (and this observation also applies to other examples which for the purpose of brevity are not included here), for the signals that include higher harmonics (sharper edges), the resulting PR-SNDR is higher, that is, the error between the Nyquist rate sampled signal and the SDS reconstructed signal is lower. For the smoother signals such as sinusoidal signals, this error would be higher unless the $\varepsilon \times T_s$ is set to a lower value which means lower compression factor. For noisy signals, the input SNRs are 23.4 dB and 20.1 dB for sine wave and saw-tooth signals, respectively and the results are plotted in Fig. 18(a) and Fig. 18(b). As one would expect, the CF for noisy input signals is lower than that of the cleaner input signal because of additional triggers of the ADC due to noise. It should be noted that the maximum achievable PR-SNDR would be the SNR of the input signals.

A. Effect of Configuration Parameters on PPG Signal

In practice, typically the general frequency- and time-domain characteristics of the signal that is being sampled, such as its frequency band and the shape of its spectrum in the frequency domain, and maximum and minimum rates of change (slopes) of the signal in the time domain are known and based on them and the amount of added error due to SDS that is tolerable, and Eqs. (1) and (13) one can estimate and adjust the values for ε and N . To further optimize ε and N for a given class of signals, one can run simulations on the representative signals of that class and find optimum values of ε and N that lead to acceptable fidelity while offering maximum compression ratio. As an example, Fig. 18(c)-18(d) depict an original PPG signal (in black), and its corresponding reconstructed signals (in red) obtained through proposed SDS technique. In this experiment, $f_{CLK} = 250$ Hz, $\varepsilon \times T_s = 7$ or 13 mV, and $N = 32$, or 64 . The difference between the reconstructed signal and original input signal, i.e., the reconstruction error is also shown with the blue curve in each case.

For the case of $N = 32$ and $\varepsilon \times T_s = 13$ mV, using the proposed SDS technique, the average number of retained samples is about 35 over a period of 5 seconds. Considering the number of discarded samples between every two retained samples that we also need to keep, to reconstruct the original signal we need to keep about 70 data samples per each 5 seconds. This is in contrast to the nominal 1250 samples that are taken with a clock frequency of 250 Hz (which is typical sampling rate for PPG applications) and using uniform sampling over a period of 5 seconds. The CF, which is defined as the ratio of the number of samples in Nyquist rate mode to the number of samples in SDS mode ($CF = \frac{N_{Nyq}}{N_{SDS}}$), in this example is ~ 17.7 .

Studying all examples, we observe that for large values of N and ε , the CF is high. In contrast, as expected, when the CF is high the PR-SNDR, which is defined as follows:

$$PR-SNDR = 10 \log \left(\frac{P(x_i - \text{mean}(x_i))}{P(e_i)} \right) \quad (19)$$

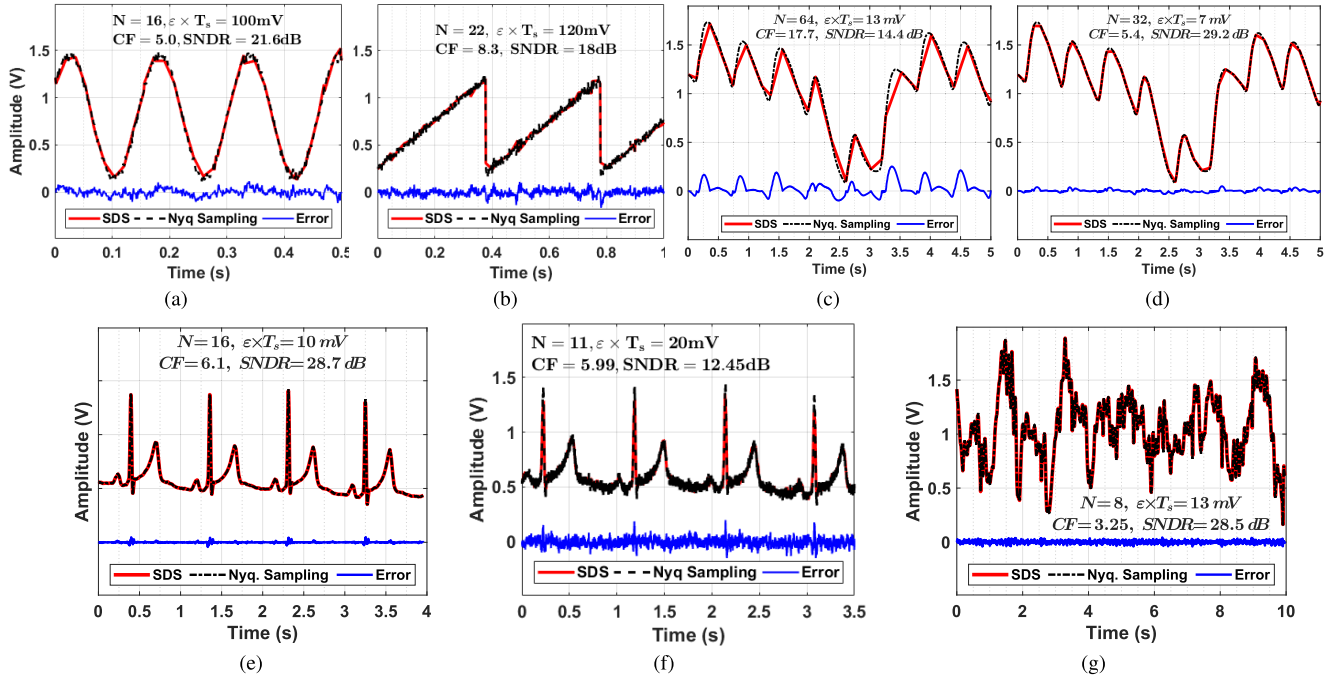


Fig. 18. Visual demonstration of the quality of the reconstructed signal and its fidelity to the original signal. Sub-figures (a) and (b) show performance for low-SNR sine wave and saw-tooth respectively, (c) and (d) show the trade-off between the signal fidelity and compression factor for PPG signal. e) performance of SDS for ECG signal. f) performance of SDS for a low-SNR ECG signal. g) performance of SDS for EEG signal.

is low. For lower values of N and/or ε the error decreases. As such, as can be seen in 18(a)-18(d), the reconstructed signal approaches to the signal sampled at the Nyquist rate.

B. Other Examples

Fig. 18(e) depicts an original ECG signal (in black), and its corresponding reconstructed signals (in red) obtained using the proposed SDS technique with $f_{CLK} = 1000$ Hz, $\varepsilon \times T_s = 10$ mV, and $N = 16$. The error signal, is also shown with the blue curve. The CF in this case is 6.1 and PR-SNDR is 28.7 dB. Fig. 18(f) also shows the results for ECG signal with low SNR of 14.8 dB. For noisy signals, to prevent the system from additional ADC triggers due to noise, based on 18, one can use a higher value of ε and a lower value for N . In this particular, where $f_{CLK} = 1000$ Hz, we have used $\varepsilon \times T_s = 20$ mV, and $N = 11$ and have achieved CF of 5.99 and PR-SNDR of 12.45 dB. For the example EEG signal, as shown in Fig. 18(g), we achieved a CF of 3.25 and a PR-SNDR of 28.5 dB by using the following setup; $N = 8$, and $\varepsilon \times T_s = 13$ mV and $f_{CLK} = 500$ Hz.

C. System-Level Power Savings

We note that in addition to reduction in the power consumption of the ADC, another advantage of the proposed approach is the additional reductions in the power consumption of the subsequent stages of the system (due to fewer data to be transferred and/or processed) and thus the reduction of the overall power of the system. In wireless and wearable sensory systems, the power consumption of the RF front-end and digital blocks of the system is usually dominant and usually is more than 95% of the overall system power [27], [29]. For example, in [29], when adaptive sampling is disabled,

TABLE II
MEASUREMENT RESULTS FOR VARIOUS SIGNALS

Signal	Nyq. Samp. Rate (S/Sec)	SDS Samp. Rate (S/Sec)	N	$\varepsilon \times T_s$ (mV)	PR-SNDR (dB)	CF	PSF (%)
PPG	250	10	64	20	9.9	24	91
PPG	250	14	64	13	14.4	17.7	89
PPG	250	16	32	20	19.9	15.1	88
PPG	250	46	32	7	29.2	5.46	76
ECG	1000	164	16	10	28.7	6.1	81
EEG	500	154	8	13	28.5	3.25	64

the power consumption of the digital and RF front-end is around $1700 \mu\text{W}$ which is more than 97% of the overall power. By using adaptive sampling which reduces the number of samples taken from the signal the overall power consumption of the system is reduced by a factor of ~ 6 . In systems in which the RF transmitter has higher output power to cover a reasonable range, the power consumption of the RF transceiver will be even more dominant. Especially, when the system has a protocol for wireless connection which forces to send and receive larger packet sizes due to handshaking and protocol stack. In such systems the power consumption during start-up time from sleep mode can not be neglected [30]. In general, the total power consumption of the system, P_{tot} , can be written as a sum of a constant portion P_{const} (that is independent of sampling rate, for example, power consumption of the instrumentation amplifier), and a data-dependent portion, P_{var} , which depends on the amount of information that should be processed and transmitted (this portion is proportional to average sampling rate). Hence,

$$P_{tot} = P_{const} + P_{var} \quad (20)$$

P_{var} can be written as $F_s \times E_{avg}$ where F_s is the average sampling rate and E_{avg} is average energy per sample for the

TABLE III
PERFORMANCE SUMMARY AND COMPARISON WITH SIMILAR WORKS

	[5]	[19]	[27]	[28]	[29]	This Work
Technique/Architecture	LC-ADC	LC-ADC	CS-AFE (RMPI)	Data Compression	Adaptive Rate	SDS+ADC
Implementation	Analog	Analog	Analog	Digital	Analog	Analog
Signal Type	All Signals	All Signals	Bio-Signals (ECG, PPG, etc.)	Bio-Signals	Only ECG	All Signals
CF (for ECG)	NA	NA	1-16	2.38	1-7.3	1-26
PSF (for ECG)	NA	NA	NA	43%	77%	81% (CF=6.1)
ADC Resolution (Bits)	NA	8	10	10	11	12
VDD	3.3 V	0.8 V	0.9 V	1 V	2 V	2 V
Power Consumption	10.7 μ W	3-9 μ W	1.8 μ W	170 μ W	8.4 μ W	1.7 μW
Nominal S.R.(for ECG)	NA	NA	1 kHz	256 Hz	1 kHz	1 kHz
CMOS Technology	0.5- μ m	0.13- μ m	0.13- μ m	65-nm	0.5- μ m	0.18-μm

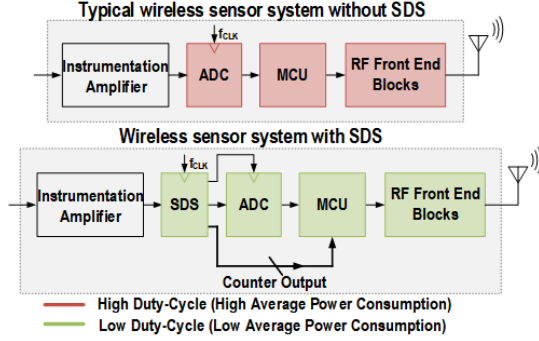


Fig. 19. System block diagram with and without SDS.

input signal. Note that the proposed SDS approach would mainly affect P_{var} and thus the proposed technique will reduce the overall power consumption P_{tot} by a factor that is slightly lower than CF. We have implemented a wireless sensor system to study the effects of SDS on power consumption of the overall system. Fig. 19 shows the block diagram of the system with and without SDS. The system uses an ultra-low-power RF microcontroller from Texas Instruments (TI), CC2650, to provide Bluetooth Low Energy (BLE) connectivity. When SDS is off, ADC is invoked at every sampling clock and converts the sensor value to the digital sample. By turning SDS on, the number of times that the ADC is invoked reduces and the number of micro-controller wake-ups and data transmissions are consequently reduced. Table II shows the results of the proposed SDS technique on various types of signals including ECG, PPG, and EEG. The user-defined settings for N and ε for each case are included in the table. Table II also reports PR-SNDR of the reconstructed signal as well as the corresponding compression factor. The last column is the Power Saving Factor (PSF), that is defined in (21).

$$PSF = (1 - \frac{P_{SDS-ON}}{P_{SDS-OFF}}) \times 100 \quad (21)$$

As it can be seen from the table, by invoking SDS depending on the signal, a power saving of at least 64% and up to 92% is achieved. The first 4 rows of the table show the dependence of PR-SNDR, CF, and PSF to the values of N and ε .

D. Discussion

Table. III shows a comparison with most relevant State-of-the-Art (SoA) designs. The proposed architecture compared favorably with the prior arts in terms of CF, PSF, and power

consumption. Note that the signal-dependent nature of the proposed technique results in a variable compression and power reduction factors. Here, to have a fair comparison, the average compression and power reduction factors are computed for the ECG signals. Furthermore, as it can be seen from Table. III, the proposed method can achieve a CF of up to 26 for ECG signals, which is 1.6-to-3.6 times better than other comparable works.

VI. CONCLUSION

In this work, we have proposed a signal-dependent sampling method and its associated circuitry for (almost) real-time sampled-data systems. We have discussed and presented closed-form analysis of the proposed method and its bounds for error and noise. For circuit implementation, we have presented several techniques to minimize the power consumption as well as the effect of nonidealities. A proof-of-concept chip is fabricated in a 0.18- μ m CMOS technology, and the performance of the system with an external ADC of the user choice is evaluated using test signals as well as real-world signals. Our experiments show that for some applications a CF of up to 16 can be achieved without a significant negative effect on the reconstructed signal, which could lead to more than 85% saving in power consumption of the overall system. In general, the performance of the proposed system compares favorably with that of the relevant SoA systems in terms of power, CF, and PR-SNDR.

ACKNOWLEDGMENT

Access to technology and chip fabrication are facilitated by CMC Micorsystems. The authors would also like to thank Rozhan Rabbani and Zohreh Azizi for their technical assistance.

REFERENCES

- [1] J. Zhou *et al.*, "Compressed level crossing sampling for ultra-low power IoT devices," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 9, pp. 2495–2507, Sep. 2017.
- [2] J. Xiang, Y. Dong, X. Xue, and H. Xiong, "Electronics of a wearable ECG with level crossing sampling and human body communication," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 1, pp. 68–79, Feb. 2019.
- [3] T. Chen, H. Kuo, and A. Wu, "A 232–1996-kS/s robust compressive sensing reconstruction engine for real-time physiological signals monitoring," *IEEE J. Solid-State Circuits*, vol. 54, no. 1, pp. 307–317, Jan. 2019.
- [4] T.-F. Wu and M. S.-W. Chen, "A noise-shaped VCO-based nonuniform sampling ADC with phase-domain level crossing," *IEEE J. Solid-State Circuits*, vol. 54, no. 3, pp. 623–635, Mar. 2019.

- [5] W. Tang *et al.*, "Continuous time level crossing sampling ADC for biopotential recording systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 6, pp. 1407–1418, Jun. 2013.
- [6] H. Wang, F. Schembari, M. Miskowicz, and R. B. Staszewski, "An adaptive-resolution quasi-level-crossing-sampling ADC based on residue quantization in 28-nm CMOS," *IEEE Solid-State Circuits Lett.*, vol. 1, no. 8, pp. 178–181, Aug. 2018.
- [7] C. Crispin-Bailey, C. Dai, and J. Austin, "A 65-nm CMOS lossless bio-signal compression circuit with 250 FemtoJoule performance per bit," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 5, pp. 1087–1100, Oct. 2019.
- [8] T.-F. Wu, C.-R. Ho, and M. S.-W. Chen, "A flash-based non-uniform sampling ADC with hybrid quantization enabling digital anti-aliasing filter," *IEEE J. Solid-State Circuits*, vol. 52, no. 9, pp. 2335–2349, Sep. 2017.
- [9] W. Guo, Y. Kim, A. H. Tewfik, and N. Sun, "A fully passive compressive sensing SAR ADC for low-power wireless sensors," *IEEE J. Solid-State Circuits*, vol. 52, no. 8, pp. 2154–2167, Aug. 2017.
- [10] T. Moy *et al.*, "An EEG acquisition and biomarker-extraction system using low-noise-amplifier and compressive-sensing circuits based on flexible, thin-film electronics," *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 309–321, Jan. 2017.
- [11] D. E. Bellasi and L. Benini, "Energy-efficiency analysis of analog and digital compressive sensing in wireless sensors," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 11, pp. 2718–2729, Nov. 2015.
- [12] Y. Tsividis, "Event-driven data acquisition and continuous-time digital signal processing," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2010, pp. 1–8.
- [13] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
- [14] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.
- [15] F. A. Marvasti, *Nonuniform Sampling: Theory and Practice*. Norwell, MA, USA: Kluwer, 2001.
- [16] H. Rauhut, "Compressive sensing and structured random matrices," *Theor. Found. Numer. Methods Sparse Recovery*, vol. 9, pp. 1–92, Jan. 2010.
- [17] M. Wakin *et al.*, "A nonuniform sampler for wideband spectrally-sparse environments," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 2, no. 3, pp. 516–529, Sep. 2012.
- [18] M. A. Davenport, J. N. Laska, J. R. Treichler, and R. G. Baraniuk, "The pros and cons of compressive sensing for wideband signal acquisition: Noise folding versus dynamic range," *IEEE Trans. Signal Process.*, vol. 60, no. 9, pp. 4628–4642, Sep. 2012.
- [19] C. Weltin-Wu and Y. Tsividis, "An event-driven clockless level-crossing ADC with signal-dependent adaptive resolution," *IEEE J. Solid-State Circuits*, vol. 48, no. 9, pp. 2180–2190, Sep. 2013.
- [20] J. Mark and T. Todd, "A nonuniform sampling approach to data compression," *IEEE Trans. Commun.*, vol. 29, no. 1, pp. 24–32, Jan. 1981.
- [21] N. Sayiner, H. V. Sorensen, and T. R. Viswanathan, "A level-crossing sampling scheme for A/D conversion," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 43, no. 4, pp. 335–339, Apr. 1996.
- [22] G. Strang, *Introduction to Linear Algebra*, 5th ed. Cambridge, MA, USA: Wellesley-Cambridge, 2016.
- [23] E. Hadizadeh Hafshejani, A. Fotowat Ahmady, and K. Anvari, "Nonuniform sampling," U.S. Patent 10333541 B1, Jun. 25, 2019.
- [24] E. Hadizadeh Hafshejani, A. Fotowat Ahmady, and K. Anvari, "Nonuniform sampling implementation," U.S. Patent 10270461 B1, Apr. 23, 2019.
- [25] M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Process. Mag.*, vol. 16, no. 6, pp. 22–38, Nov. 1999.
- [26] B. Razavi, *Design of Analog CMOS Integrated Circuits*, 2nd ed. New York, NY, USA: McGraw-Hill, 2016.
- [27] D. Gangopadhyay, E. G. Allstot, A. M. R. Dixon, K. Natarajan, S. Gupta, and D. J. Allstot, "Compressed sensing analog front-end for bio-sensor applications," *IEEE J. Solid-State Circuits*, vol. 49, no. 2, pp. 426–438, Feb. 2014.
- [28] E. Chua and W.-C. Fang, "Mixed bio-signal lossless data compressor for portable brain-heart monitoring systems," *IEEE Trans. Consum. Electron.*, vol. 57, no. 1, pp. 267–273, Feb. 2011.
- [29] R. F. Yazicioglu, S. Kim, T. Torfs, H. Kim, and C. Van Hoof, "A 30 μ W analog signal processor ASIC for portable biopotential signal monitoring," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 209–223, Jan. 2011.
- [30] D. Griffith, J. Murdock, and P. T. R  ne, "5.9 A 24 MHz crystal oscillator with robust fast start-up using dithered injection," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Jan. 2016, pp. 104–105.



Ehsan Hadizadeh Hafshejani received the B.S. degree in electrical engineering from Shahrekord University in 2010 and the M.S. degree from the Sharif University of Technology in 2013, where he is currently pursuing the Ph.D. degree. He is also a Visiting Research Scholar with the Electrical and Computer Engineering Department, The University of British Columbia. His research interests include RF/Analog IC design and ultra-low power systems.



Mohammad Elmi received the B.S. degree in electrical engineering from the Babol Noshirvani University of Technology, Mazandaran, Iran, in 2013, and the M.S. degree in electrical engineering-analog electronics from Shahid Beheshti University, Tehran, Iran, in 2015. His current research interests include analog/RF integrated circuits design, low-power biomedical circuits design, and mm-wave integrated circuits.



Nima TaheriNejad (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from The University of British Columbia (UBC), Vancouver, Canada, in 2015. He is currently an Assistant Professor with TU Wien (formerly known as Vienna University of Technology as well), Vienna, Austria. His current research interests include self-awareness in resource-constrained cyber-physical systems, embedded systems, systems on chip, memristor-based circuit and systems, health-care, and robotics.



Ali Fotowat-Ahmady (Member, IEEE) was born in Tehran, Iran, in 1958. He received the B.S. degree from the California Institute of Technology, Pasadena, CA, USA, in 1980, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 1982 and 1991, respectively. He started his career at Philips Semiconductor, Sunnyvale, CA, USA, in 1987, where he developed several integrated circuits for mobile phones. Since 1992, he has been with the Department of Electrical Engineering, Sharif University of Technology, Iran, where he is currently an Associate Professor.



Shahriar Mirabbasi (Member, IEEE) received the B.Sc. degree in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 1990, and the M.A.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 1997 and 2002, respectively. Since August 2002, he has been with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, Canada, where he is currently a Professor. His current research interests include analog, mixed-signal, RF, and mm-wave integrated circuit and system design with an emphasis on communication, sensor interface, and biomedical applications.