2013

Analog Signal Buffering and Reconstruction

Brandon M. Kelly
West Virginia University

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

Recommended Citation

Kelly, Brandon M., "Analog Signal Buffering and Reconstruction" (2013). *Graduate Theses, Dissertations, and Problem Reports*. 3612.

<https://researchrepository.wvu.edu/etd/3612>

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

Analog Signal Buffering and Reconstruction

by

Brandon M. Kelly

Thesis submitted to the
Benjamin M. Statler College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Electrical Engineering

David W. Graham, Ph.D., Chair
Matthew Valenti, Ph.D.
Vinod Kulathumani, Ph.D.

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2013

Keywords: Analog, Integrated Circuits, Wireless Sensor Networks, Adaptive Sampling,
Field-Programmable Analog Arrays

Copyright 2013 Brandon M. Kelly

Abstract

Analog Signal Buffering and Reconstruction

by

Brandon M. Kelly

Master of Science in Electrical Engineering

West Virginia University

David W. Graham, Ph.D., Chair

Wireless sensor networks (WSNs) are capable of a myriad of tasks, from monitoring critical infrastructure such as bridges to monitoring a person's vital signs in biomedical applications. However, their deployment is impractical for many applications due to their limited power budget. Sleep states are one method used to conserve power in resource-constrained systems, but they necessitate a wake-up circuit for detecting unpredictable events. In conventional wake-up-based systems, all information preceding a wake-up event will be forfeited. To avoid this data loss, it is necessary to include a buffer that can record prelude information without sacrificing the power savings garnered by the active use of sleep states.

Unfortunately, traditional memory buffer systems utilize digital electronics which are costly in terms of power. Instead of operating in the target signal's native analog environment, a digital buffer must first expend a great deal of energy to convert the signal into a digital signal. This issue is further compounded by the use of traditional Nyquist sampling which does not adapt to the characteristics of a dynamically changing signal. These characteristics reveal why a digital buffer is not an appropriate choice for a WSN or other resource-constrained system.

This thesis documents the development of an analog pre-processing block that buffers an incoming signal using a new method of sampling. This method requires sampling only local maxima and minima (both amplitude and time), effectively approximating the instantaneous Nyquist rate throughout a time-varying signal. The use of this sampling method along with ultra-low-power analog electronics enables the entire system to operate in the μW power levels. In addition to these power saving techniques, a reconfigurable architecture will be explored as infrastructure for this system. This reconfigurable architecture will also be leveraged to explore wake-up circuits that can be used in parallel with the buffer system.

Analog Signal Buffering and Reconstruction

by

Brandon M. Kelly

Thesis submitted to the
Benjamin M. Statler College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

Lane Department of Computer Science and Electrical Engineering

APPROVAL OF THE EXAMINING COMMITTEE

Matthew Valenti, Ph.D.

Vinod Kulathumani, Ph.D.

David W. Graham, Ph.D., Chair

Date

Acknowledgments

I would like to begin by thanking my adviser, Dr. David W. Graham, for the opportunity to perform this research and for his constant guidance and support. I would also like to thank my committee for the support they have provided me in my graduate and undergraduate studies. Brandon Rumberg also deserves a special thanks for his patience and guidance throughout the entirety of my research.

Finally, I would like to thank my family and friends who have supported me beyond measure. Whenever I have needed some help or just a kind word of encouragement, I have never had far to look. I am truly a fortunate man to have so many thoughtful and caring people in my life.

Contents

Approval Page	i
Abstract	ii
Acknowledgments	iii
List of Figures	vi
1 Introduction	1
1.1 Outline	3
2 Challenges in Wireless Sensor Networks	4
2.1 Power Saving Schemes in WSNs	5
2.2 Analog Electronics and Analog Processing	6
2.3 Minimal Sampling Techniques	7
2.4 Chapter Summary	8
3 Theory of Peak Sampling and Reconstruction	10
3.1 Peak Sampling	12
3.2 Bézier Reconstruction	14
3.3 Reconciliation with Digitally Sampled Information	18
3.4 Disadvantages of Peak Sampling	19
3.5 Chapter Summary	20
4 Analog Buffer Circuits	21
4.1 System Overview	22
4.2 Operational Transconductance Amplifier	24
4.3 Peak Detector	26
4.3.1 Track-and-Hold	28
4.3.2 Crossing Style	30
4.4 Short-Term Memory	33
4.5 Timing Circuit	35
4.6 System Examples	37
4.7 Conclusions	40

5	Reconfigurable Infrastructure for Analog Circuitry	41
5.1	FPAA Architecture	42
5.2	Mote Interfacing	44
5.2.1	User Interface	46
5.2.2	Compression	47
5.3	System Examples	49
5.4	Conclusions	52
6	Conclusion	53
	References	56

List of Figures

1.1	To avoid the data loss normally associated with sleep states in WSNs, we introduce an ultra-low-power analog memory buffer to operate in parallel with a wake-up circuit. This system records data from the sensor while the WSN mote is allowed to remain in a sleep state, enabling signal approximation without sacrificing the energy savings from the sleep state.	2
3.1	Our sampling method only samples the local maximums and minimums of a signal.	13
3.2	Peak sampling yields savings in samples taken when compared to traditional Nyquist sampling.	14
3.3	An example of a section of a wave recreated using the Bézier curve formula .	16
3.4	(a) A speech waveform and its corresponding peak sample points. (b) A speech waveform recreated using peak sample points. Qualitatively, the reconstructed waveform is intelligible and sounds like a filtered version of the original waveform. This filtered sound corresponds with the shortcoming of the system, which are described in the next section.	17
3.5	A section of the speech waveform that was reconstructed in the previous figure. The Bézier reconstruction method was applied to the peak samples of the original waveform. The actual reconstruction was then resampled at 15.8kHz, as would be required by a traditional system for this particular waveform. . .	18
3.6	An example of a change in the derivative between two peaks that does not generate extra local maximums or minimums. This information is forfeited by peak sampling.	19
4.1	To avoid the data loss normally associated with sleep states in WSNs, we introduce an ultra-low-power analog memory buffer to operate in parallel with a wake-up circuit. This system records data from the sensor while the WSN mote is allowed to remain in a sleep state, enabling signal approximation without sacrificing the energy savings from the sleep state.	22
4.2	Our analog memory buffer locates the local maxima and minima in real time and stores their respective amplitudes and times separately.	23
4.3	A basic operational transconductance amplifier.	24
4.4	A sweep of the output current versus the differential input voltage for a typical operational transconductance amplifier.	25

4.5	A more advanced operational transconductance amplifier.	25
4.6	The symbol used to denote operational transconductance amplifiers throughout the rest of this work. The specific OTA referred to is the nine transistor version.	26
4.7	The maximum locator circuit consists of an envelope detector, a comparator, and a pulse-generation circuit. The envelope detector tracks the input on the rising slope of the signal and then slowly decays on the falling slope of the input signal. This tempered decay rate causes the comparator to detect a discrepancy at the point when the signal begins to decay, i.e. the local maximum. The comparator then signals a pulse generator, which triggers the write phase in the memory storage unit. The minimum locator circuit operates similarly, tracking the input on the falling slope and lagging the input on the rising slope.	27
4.8	The max/min locator circuits operate by tracking the input on the rising/falling slope, and then decaying at a slower rate. The discrepancy between the decay of the max/min locator circuit and the original signal is detected by the comparator, which in turn signals the pulse generator to activate a write pulse. These data were taken from a chip fabricated in a $0.5\mu\text{m}$ standard CMOS process available through MOSIS.	28
4.9	The track-and-hold style peak detector, with respect again to the maximum locator, consists of a rising edge tracker, a comparator, a pulse generation circuit, and a transmission gate controlled by the minimum locator's pulse generation circuit. Essentially, the device will track rising edge of the input signal in an attempt to detect local maximums and will reset to the input during the only time that one can be certain that a maximum has not occurred and is not presently occurring, during a minimum.	29
4.10	Simulated output of the track-and-hold style peak detector. The input wave is traced by a maximum locator circuit. The minimum pulse generator causes the maximum locator to reset to input. The minimum locator circuit behaves similarly.	30
4.11	Simulated output of a crossing style peak detector. The output wave crosses the input at only the local maximums and minimums. This sharp transition triggers the comparator to create an upward edge on the minimums and a falling edge on the maximums.	31
4.12	A crossing style peak detector constructed with a single OTA	31
4.13	A crossing style peak detector constructed with two OTAs	32
4.14	A crossing style peak detector constructed with a single self-biasing OTA	33
4.15	The capacitor bank tracks the system input and enters write operation only when the max/min locator's pulse generator outputs a logic high signal. After each write, the address of the selected capacitor is incremented through the break-before-make multiplexer.	34
4.16	The timer is a basic capacitive time-to-voltage converter. A bias current linearly discharges a capacitor. The charge on the capacitor can be captured, stored, and then easily processed into a time value utilizing the known rate of discharge. The capacitor is reset after every sample.	35

4.17	Simulated output of the timer. The output waveform is the result of the linear discharge of the capacitor.	36
4.18	The system is capable of capturing a wide range of amplitudes and frequencies. For these plots, the input is shown in grey, detected max/mins are black dots, and the reconstructed wave is a black dashed line. These data were taken from a chip fabricated in a $0.5\mu\text{m}$ standard CMOS process available through MOSIS.	38
4.19	(a) The speech clip used as input to the system is shown in grey with detected max/mins in black dots. (b) The speech clip approximation from using the modified Bézier formula with the sampled max/mins. These data were taken from a chip fabricated in a $0.5\mu\text{m}$ standard CMOS process available through MOSIS.	39
5.1	A sensor node equipped with a field-programmable analog array (FPAA). The FPAA can be reconfigured in run-time to perform event detection and pre-processing at a power consumption that is significantly lower than the power that would be consumed by the mote's built-in digital systems.	42
5.2	Die photograph of the FPAA fabricated in a standard $0.5\mu\text{m}$ CMOS process available through MOSIS. The FPAA is 2.25mm^2	43
5.3	PCB for interfacing the FPAA to a WSN sensor node. This board incorporates a variety of sensors, a CPLD, a DAC (on the underside of the PCB), and a socket for connecting a TelosB mote.	45
5.4	A high-level schematic of the PCB for interfacing sensors, the FPAA, and a WSN mote.	45
5.5	(a) Example of a configuration of switches. (b) Implementation of how this configuration would be transmitted using our compression scheme.	47
5.6	Top-level schematic of the circuit synthesized in the FPAA to demonstrate its spectral analysis capabilities. The circuit detects portions of the signal where the frequency content rises in the 2-4kHz range. The "Correlation" stage detects the simultaneous presence of content in the high-frequency band and the delayed low-frequency band. The "Inhibition" stage nulls the output when content is present in the low-frequency band to avoid triggering on wideband signals. G_{m2} is biased by the gate of the attached pFET while the output current of G_{m1} is mirrored through the diode connected FET. Also note that $V_{pulldown}$ is a constant bias used to weakly pull down the output when G_{m2} is shutoff by G_{m1}	48
5.7	Spectral analysis performed by the analog IC. (Top, Middle) The transient plot and the spectrogram plot of the input signal, respectively. The sinusoidal signal, which includes Gaussian noise, varies from 1kHz to 8kHz and concludes with a steady input of ten sine waves ranging from 2kHz to 4kHz. (Bottom) The output stage successfully detects portions of the signal where the frequency content rises in the 2-4kHz range.	49
5.8	Schematic of the implemented voice-activity detection algorithm. The device triggers an event when the amplitude modulation in the speech band occurs at a rate that is typical of speech.	50

5.9	Demonstration of the analysis of a signal throughout the voice-activity detection algorithm. For this test, the input was a male voice corrupted by noise from an airport environment, at a signal-to-noise ratio of 10dB.	51
5.10	A block diagram showing the implementation of the peak detection algorithm in conjunction with a gyroscope.	52

Chapter 1

Introduction

Many of the technologies we look forward to today, including automation of simple tasks or increased safety from passive machine monitoring, rely on the premise that we will have access to small computing platforms capable of processing data from a variety of sensors. Wireless sensor networks (WSNs) are a current technology that fit such a description, but their proliferation is limited by their restrictive power budgets. Fortunately, there are technologies and techniques which hold promise in increasing the lifetimes of WSNs.

One such technology falls under the broad category of wake-up circuits. A wake-up circuit allows the WSN node to remain in a low-power sleep state until an event occurs that must be monitored. This event can either trigger some threshold within the wake-up circuit, or the circuit itself could act as a timer, periodically waking the system up. However, even if the wake-up circuit is ideal and triggers on every event, and even if it wakes up the system in a short enough time that the beginning of the event is not missed, it is still consciously forfeiting information about the prelude to the event. For some applications, data about the prelude of an event could be useful.

Unfortunately, the use of a digital memory buffer would likely offset the majority of energy savings garnered by the use of sleep states. A digital memory buffer would require costly conversions of real-world analog signals into digital signals. However, if a buffer were based on analog circuitry, it would be able to operate in the analog domain without having to expend energy on conversion. By avoiding this conversion, the system would be able to operate in parallel with the wake-up circuit and retain the energy savings garnered by the

Figure 1.1: To avoid the data loss normally associated with sleep states in WSNs, we introduce an ultra-low-power analog memory buffer to operate in parallel with a wake-up circuit. This system records data from the sensor while the WSN mote is allowed to remain in a sleep state, enabling signal approximation without sacrificing the energy savings from the sleep state.

use of sleep states (Fig. 1.1). The energy consumption of the buffer could be even further reduced if it were made to sample at an adaptive rate, thus always staying on the edge of the Nyquist rate for a variety of dynamic signals.

Such a device would help to alleviate the drawbacks of sleep states and would help enable the proliferation of WSNs, but people would likely still prefer digital systems. Digital systems are easily scaled to a given task and easily programmed or configured through reconfigurable architectures such as field programmable gate arrays. These architectures make prototyping and device deployment simple by allowing users to program the devices in a manner similar to software. Therefore, for the analog memory buffer to actually be adopted by WSN architects, a simple and reconfigurable platform should also be developed through which it can be implemented.

Such a device would help to alleviate the drawbacks of sleep states, but it would ultimately be application-specific. Like many analog solutions, this device would have to be designed with a certain range of applications in mind so that it could be tuned for the target signals'

characteristics. Conversely, digital systems can be built upon reconfigurable architectures which allow the system to be tuned for a given application post-fabrication. Adding this reconfigurability would greatly increase the viability of an analog-based system.

The goal of this work is to address all of the preceding concerns:

- To develop an ultra-low-power analog memory buffer which can be used in parallel with a wake-up circuit in a WSN node.
- To utilize a sampling scheme that adapts to a signal's changing characteristics, thus maximizing system resources.
- To begin to develop a reprogrammable infrastructure through which the entire analog front-end, both buffer and wake-up circuit, can be programmed.

1.1 Outline

Chapter 2 will provide a brief overview of fields related to this work including WSN energy saving schemes, analog electronics and signal processing, as well as a brief primer on minimal sampling. Chapter 3 will document a novel sampling method by which an analog system can maintain a minimal sampling rate on a signal of changing frequency content. Chapter 4 will then detail the development of an analog buffer which takes full advantage of the previously mentioned sampling rate. Chapter 5 will begin to answer how such a buffer could be practically implemented by exploring a reconfigurable architecture which it and the wake-up circuit can utilize. Finally, Chapter 6 will conclude this work and suggest future directions for this research.

Chapter 2

Challenges in Wireless Sensor Networks

Wireless sensor networks (WSNs) are discrete computing platforms that show great promise for a variety of applications. These applications range from biomedical applications such as body sensor networks, to infrastructure monitoring such as monitoring bridge integrity, and even monitoring and tracking animals based upon their vocalizations [1, 2]. WSNs represent the next important ingredient in making many imagined cyber-physical systems a reality, yet the deployment of WSNs has been restrained.

The slow adoption of WSN technology is attributed to the limited power budget. Complex and exotic computational tasks, such as those previously mentioned, require the use of energy-draining digital circuitry. Further power consumption occurs from superfluous in-network communication which is often required to simply deem whether data is relevant or irrelevant.

This chapter will outline various schemes used to overcome the limited power budgets of WSNs. After demonstrating their relative strengths and shortcomings, a brief discussion of analog processing and non-Nyquist based sampling techniques will be presented which will serve as a primer to the solution presented throughout the rest of this work.

2.1 Power Saving Schemes in WSNs

Power budgets are a primary concern within WSNs [3]. In the most general terms, there are two schools of thought on how to reduce energy expenditure for individual WSN platforms, called nodes. The first viewpoint is that the computational load of a node should be reduced by having it broadcast raw data to a more capable base station or to other nodes for distributed processing. Unfortunately, this method can add considerable communication overhead. As radio usage and other communication protocols are a primary source of energy drain to begin with, this method is not optimal.

Still, there are many notable methods which attempt to mitigate this extra communication overhead cost. Methods of reducing communication energy by improving communication protocols include minimum energy transmission protocols, direct transmission, multi-hop routing, static clustering and dynamic clustering methods [4]. While many of these schemes require higher energy expenditure in terms of local processing, their cost is more than offset by the savings garnered from reducing communication overhead. A plethora of these techniques exist, but most of their end goals can be simplified to placing an equal burden across a sensor network, ensuring that communication duties are fairly distributed and one overtaxed node does not cause the entire network to fail prematurely [5, 6].

The other school of thought is to reduce the amount of information that needs to be transmitted by performing a majority of the computations locally, at each individual node. This scheme of course saves on communication, but requires more energy for processing, and requires a platform actually capable of performing the necessary computations - a non-trivial requirement. While this approach has its drawbacks, there are means of mitigating these drawbacks which makes this our preferred approach.

The primary means of mitigating the energy requirements of digital processing is to utilize said resources as little as possible. In this scheme, the digital portions are left in a low-power sleep state while some other wake-up circuit runs. The wake-up circuit can be periodic or event driven. Either method creates a great deal of energy savings inversely proportional to the node's duty cycle [7]. However, care must be taken to ensure that valuable data from events are not missed.

Wake-up based detectors are of great interest to the WSN community. Dutta, et al. developed an extreme scale WSN equipped with infrared and acoustic sensors as part of the system's wake-up circuitry [8]. Jevtic, et al. reported in [9] a WSN for event-driven applications which achieved $1/245$ the power consumption of a comparable non-wake-up based WSN. A WSN designed for cargo monitoring was developed by Paradiso, et al. [10] which consumes an average of only $25\mu\text{W}$ of power. Goldberg, et al. [11] reported on a wake-up detector which monitors for periodicity in low-frequency audio signals. These examples are just a minor sampling of some of the contributions currently being made to the field of wake-up detector equipped WSNs.

As promising as this path may be, wake-up based nodes face a couple common drawbacks that cannot be overcome without some external solution. Primarily, even in an ideal case where the node is awake for every event, there is still some data-loss associated with the wake-up time. If, however, we also are assuming ideal nodes that can wake-up in a negligible amount of time, we are still forfeiting valuable information about the prelude. Recovering this valuable prelude information without forfeiting the energy savings garnered by sleep states is achieved through the use of analog electronics and asynchronous sampling in this work.

2.2 Analog Electronics and Analog Processing

The stringent power requirements of WSNs lend themselves to the use of ultra-low-power analog electronics. While many computational tasks traditionally fall within the digital domain, it has been shown that many such tasks can be performed more easily in the analog domain [12, 13]. In [14], Hasler et al. presented an analog auditory sensory system that provided a power savings of 3-4 orders of magnitude over a comparable digital system. This power savings was shown to be the equivalent of a 20 year leap in digital fabrication process technology. In [15], it was shown that an analog system could be used to provide spectral analysis for a WSN at an average power consumption of $1\text{-}3\mu\text{W}$.

Due to their efficiency and natural computational capabilities, analog electronics have found a recent resurgence in many medical applications, where replacing a system's battery

is inconvenient at best or requires surgery at worst. Analog electronics have the inherent potential to perform spectral decomposition at a low power cost, thus making them a natural choice for auditory prostheses [16]. They have also found use in implantable, neural recording systems [17].

Clearly, the low power consumption of analog signal processing (ASP) is well-suited for WSN implementation. However, the quick adoption of this technology has been restrained by the fact that ASP implementation requires a priori knowledge of the application space. Also, ASP development is slow relative to its digital counterparts, largely due to the ease of design offered by reconfigurable digital systems. One solution to these challenges is to enable ASP reconfiguration through the use of a field-programmable analog array (FPAA) [18, 19, 20]. The FPAA allows architects of WSNs to rapidly prototype and test ASP designs in real-world conditions.

FPAAAs have already been proven to be capable of synthesizing complex circuitry. [18], for example, details an FPAA that was constructed with a large number of computational analog blocks and is capable of synthesizing many complicated ASP applications, including an AM receiver and an analog speech processor. While there has been considerable research performed on the construction of FPAAAs, very little has been done on their inclusion within large-scale systems such as WSNs. This work will begin to bridge that gap.

2.3 Minimal Sampling Techniques

Shannon first presented the Nyquist sampling rate in [21]. It was here that he demonstrated how sampling at twice the greatest frequency content of a band-limited signal could yield sufficient information for the reconstruction of that signal. A drawback of this method is that the system inevitably over-samples during periods of relatively low-frequency content. This oversampling is a primary source of inefficiency within sensing systems. For this reason, researchers have looked to craft more efficient methods of sampling. These methods fall generally into two categories: explicit sampling and implicit sampling.

In an explicit sampling method, the time or rate at which sampling occurs is explicitly defined. Since Shannon's original work, explicit sampling has been very well studied for many

cases, and its generalized sampling theorem has been reviewed and extended by several authors [22, 23]. Another important and well studied area is in the area of non-uniform sampling. Within the context of explicit sampling, non-uniform sampling is essential in real-world systems which may have sampling times that drift from their ideal locations [24]. While this drift is generally treated as an issue within the context of explicit sampling, samples that are intentionally non-uniformly spaced tend to fall under the category of implicit sampling.

Implicit sampling includes methods where the time or rate is not specified; sampling is instead triggered by some other quality of the signal [25]. The most famous of these is arguably Bond and Cahn's work [26]. They suggested that a band-limited signal could be adequately sampled by sampling only the zero crossings of a function. Beyond Bond and Cahn, several others have conducted important studies, including sampling based upon level crossing [27] or sampling based upon the crossings of a known function, such as a cosine function [28]. There have also been more recent developments in implicit sampling, including [29] which advocates a time-stampless method that adjusts dynamically to a signal based upon a mathematical model of accruing instantaneous frequencies.

The description of the problem of oversampling along with our method of minimal sampling may call to mind compressed sampling [30]. Compressed sampling attempts to solve the problem of oversampling by projecting the signal of interest onto some mathematically lower-dimensional domain prior to sampling. While there are some promising hardware approaches to compressive sampling [31], most are software based and carry severe power restrictions which make them inappropriate for energy-constrained applications such as WSNs [32]. Primarily, the projection of the signal onto a lower-dimension requires considerable mathematical computation and also requires that the signal first be explicitly sampled. While the resulting compressed samples may help limit communication overhead when they are transmitted, the method with which they are attained is impractical in many WSNs.

2.4 Chapter Summary

There are many approaches that have been taken to enabling the proliferation of WSNs by extending their battery life. Some deal directly with the system, as in the case of wake-up

detectors. Others attempt to modify the computational circuitry of the individual devices composing the systems, as in the case of ultra-low-power analog electronics. And finally, some methods attempt to reduce the action of the entire network through the use of minimal sampling techniques. A review of the literature of each field yields unique strengths and weaknesses, but for a complete solution, we will attempt a comprehensive approach.

The first step we have taken in reducing the energy-consumption of WSNs is to minimize the number of samples that must be taken by the system. We do so by utilizing a novel new implicit sampling technique that minimizes the required samples without adding computational complexity. Next, we take this concept and merge it with analog electronics to create a buffer which consumes a fraction of the power utilized by an active WSN mote. Finally, we make the entire system practical by exploring its implementation within the framework of an FPAA. This allows us not only to make our own buffer non-application specific, but allows us to explore the wake-up circuits that it will operate with in parallel.

Chapter 3

Theory of Peak Sampling and Reconstruction

Digital systems have an inherent difficulty in understanding the real-world. These digital systems make their computations in one's and zero's, black and white. The real-world is analog and is composed of many shades of gray. For digital systems to interact with the real-world, they must be able to convert analog signals into the digital domain.

Systems that take real-world signals from the analog domain into the digital domain consume a great deal of power simply performing that conversion. Analog signals must be sampled, transformed into digital signals using an analog to digital converter, and then either stored in memory for later use or processed by high-energy digital circuitry. Every step in this process creates a large amount of overhead that is directly proportional to the number of samples used to convert the analog signal into the digital domain.

For a wireless sensor network (WSN) or any sensor-based system to consume a minimal quantity of power, it is not only necessary to design efficient electronics, it is also necessary to analyze the method in which signals are sampled and reconstructed. Traditional sampling and reconstruction methods begin with Nyquist sampling. Nyquist sampling necessitates sampling at a frequency greater than twice the highest frequency component of the original signal. Using this constant rate of sampling leads to oversampling when the signal does not contain the highest allowable frequency components [33]. Wasting system resources on superfluous samples is a primary source of inefficiency in sensor-based systems.

A system designed for minimal power consumption should be able to adapt in real time to the changing frequency of the signal being sampled to avoid resource inefficiency. We suggest that sampling only the local maxima and minima (both amplitude and time) provides sufficient information about a time-varying signal to produce an adequate approximation. Recording only max/min values, for a properly bandlimited signal, yields a sampling rate that adjusts dynamically to the instantaneous Nyquist rate. This advantage is compounded by our knowledge of the function's first derivative at every sample point. Since each sample point is a local maximum or minimum, we know that the derivative at every sample point is equal to zero. We will later demonstrate how this extra piece of information is invaluable in the reconstruction process. It is worth mentioning first, however, that scheduled wake-up cycles are adequate for sampling slowly varying signals and DC levels, while still maintaining low-power performance. Therefore, we focused our analysis on time varying (AC) signals.

Beyond simply demonstrating the validity of our sampling theorem, a computationally simple method of reconstruction would be very useful within WSNs. Traditional reconstruction methods, aggregating sinc waves scaled to each sample point, are too complex for the resource constrained WSN nodes. Typically, this would not be a problem as samples could simply be transmitted to a more computationally capable base station for reconstruction and analysis. While this may often be the case, there are situations where local reconstruction could be very beneficial. First, local reconstruction could allow the node to perform some analysis to see if there is any information within the recorded data that is worth broadcasting at all. This would allow the node to trade some extra processing time in exchange for reduced energy expenditure from communication. Additionally, local reconstruction could allow a signal that is sampled after wake-up to be pre-pended with information from the prelude - thus yielding a complete measurement of the event. Beyond this, it could use the information it gains from reconstruction to adjust its wake-up schedule or wake-up event so that it may record more or less information using its higher-resolution digital components. This action would allow the node to self-optimize its duty cycle. These are a few of the reasons why we developed a reconstruction method which complements our minimal sampling technique.

3.1 Peak Sampling

One of the classical signal processing papers is Shannon's Communication in the Presence of Noise [21]. Here, Shannon shows that for a signal to be reconstructed completely, it must be sampled at a frequency of greater than twice its highest frequency component. This rate, known as the Nyquist rate, effectively became the standard for interpreting real-world analog signals in a digital environment.

Utilizing the Nyquist rate does carry with it some drawbacks and associated challenges. Chief among them is the challenge of using a fixed rate on a signal of widely varying frequency content. If the highest frequency component of a signal is relatively rare, then the Nyquist rate will be unnecessarily high for the majority of the duration of the signal. For example, consider a system designed for sampling over the human audible range of 20Hz to 20kHz. The minimum sampling frequency dictated by the Nyquist rate would be 40kHz. For portions of the signal that are at the 20Hz level, the Nyquist sampling rate will be one thousand times what is needed. In other words, the system will be required to sense a signal, convert it to the digital domain, and either store its value or process it one thousand times more often than is absolutely necessary. Many have tried to address this issue by devising sampling schemes that adapt to some characteristic of the signal [28, 29].

One seminal paper in the area of adaptive sampling was written by Bond and Cahn and focused on sampling the real zeros of a band-limited signal [26]. By sampling only the real zeros, which geometrically correspond to the zero-crossings of the signal, they demonstrated that they were essentially approximating the Nyquist rate. However, they did note that a deficiency in this theory stems from the work of Titchmarsh in [34]. It was here that Titchmarsh theorized that a signal could be approximated by its real and complex zeros which occur, in total, at the Nyquist rate. Bond and Cahn felt that, because there was no way of detecting complex zeros for a continuous function, it was most useful to discuss how a signal segment could be approximated in terms of real zeros only.

Our adaptive sampling method relies on sampling the local maximum and minimums, or extrema, of a signal (Fig. 3.1). Both the time and amplitude values of the peaks are sampled. By sampling a band-limited function at only the local maximums and minimums, we are

Figure 3.1: Our sampling method only samples the local maximums and minimums of a signal.

effectively sampling at the instantaneous Nyquist rate. In addition to this, our sampling method carries with it one extra piece of information, the derivative at every sample point. The knowledge of this derivative, specifically that the derivative is equal to zero, is valuable information. In fact, Shannon remarked in his original paper that a signal could also be reconstructed if the value of the signal and its first derivative were known at every other sample point [21]. We use the knowledge of this derivative to simplify the reconstruction method, as described in the following section.

This sampling method carries with it a tremendous savings over the traditional Nyquist sampling method in signals of varying frequency. Consider the sinusoidal signal of varying frequency in Fig. 3.2. Our peak sampling method will sample the waveform twice per period, or approximately the instantaneous Nyquist rate, no matter what the instantaneous frequency is throughout the signal. Traditional Nyquist sampling of the same waveform is dictated by the highest frequency component and results in over twice as many samples being taken for the same signal segment.

Figure 3.2: Peak sampling yields savings in samples taken when compared to traditional Nyquist sampling.

This method of sampling could also be linked to the previously mentioned theory devised by Titchmarsh [34]. The reason for this is that if one were to approximate the signal as a series of parabolas, then the complex zeros of each parabola would correspond geometrically to its local maximum or minimum. Therefore, peak sampling could be viewed as approaching Titchmarsh's theory from the opposite direction of Bond and Cahn, by sampling only the complex zeros of a signal.

3.2 Bézier Reconstruction

Classical sampling requires the use of the sinc function, which is the Fourier transform of the rectangular function. The rectangular function is known more colloquially as a brick-wall low-pass filter. Sample points are represented as evenly spaced pulses, spaced according to the sampling frequency and scaled to the appropriate sampled amplitude. Ideally, this infinite pulse train is then passed through the brick wall filter, which eliminates all frequency

content beyond the corner frequency of the filter. Thus, yielding a perfect reconstruction of a band-limited signal. In reality, the accuracy of this reconstruction is limited by the use of a non-ideal low-pass filter in place of a brick wall filter and is also diminished by the use of a limited number of samples. These disadvantages are typically overcome through the use of oversampling, which is impractical for a sample-limited application like a memory buffer.

These classical reconstruction methods are inappropriate for the irregularly spaced sample points gained through the use of peak sampling. Simply put, it is unclear where the corner frequencies of the brick wall filter should be set since the sampling rate is in constant flux. Beyond this, classical sampling methods would not be able to take advantage of the known derivative of the function at every sample point. Not using this knowledge in a resource constrained system, such as a memory buffer, would be wasteful.

For peak sampling to be an effective method of sampling within an energy-constrained system, it should ideally also have a simple method of reconstruction. While it would be possible to perform the reconstruction at a less energy-constrained base station, there are applications where reconstruction at the node would be useful. One hypothetical example would be where a node might use the reconstructed prelude information to adjust the timing of its wake-up event. In this manner, the node could learn when the ideal wake-up time is that balances energy conservation with observing the target event. It is for this reason that we have developed Bézier Reconstruction.

The Bézier method of reconstruction is a simple, yet powerful, interpretation of the Bézier curve formula [35] given by:

$$B(x) = (1-x)^3 P_0 + 3(1-x)^2 x P_1 + 3(1-x) x^2 P_2 + x^3 P_3, \quad x \in [0, 1] \quad (3.1)$$

The Bézier curve formula generates points along a smooth curve that is specified by the endpoints, P_0 (at $x=0$) and P_3 (at $x=1$), and concavity points, P_1 and P_2 . We utilize the formula sample-by-sample to interpolate between every pair of adjacent max/min values. Thus, we take N max/min samples and generate $N - 1$ segments between them to create a full approximation of the signal. For each segment, the formula is applied to the voltage and time values separately. By understanding that all sample values are located where the derivative of the input equals zero (i.e., where the slope is zero), the Bézier formula can be

Figure 3.3: An example of a section of a wave recreated using the Bézier curve formula

simplified by setting the points P_1 and P_2 to be equal to the max/min locations. Accordingly, the equation for interpolating the voltage between samples $k - 1$ and k is

$$V(x) = (1 - x)^3 V_k + 3(1 - x)^2 x V_k + 3(1 - x) x^2 V_{k-1} + x^3 V_{k-1} \quad (3.2)$$

The result of this equation is a nearly sinusoidal-shaped curve that spans the specified amplitudes. The time vector is interpolated similarly, except that the concavity points are set to the midpoint of the time interval $T_{kmid} = (T_k + T_{k-1})/2$, as

$$T(x) = (1 - x)^3 T_k + 3(1 - x)^2 x T_{kmid} + 3(1 - x) x^2 T_{kmid} + x^3 T_{k-1} \quad (3.3)$$

The result is a vector of time values that shift the sinusoidal curve to the appropriate time endpoints (Fig. 3.3).

A more application-driven example of this reconstruction method can be seen in Fig. 3.4. Here, we see a filtered speech waveform that represents a potential input to a sensing system. As can be seen, the local maximums and minimums were appropriately sampled and then used to provide a reconstruction. The peak sampling method in this example captured 552

Figure 3.4: (a) A speech waveform and its corresponding peak sample points. (b) A speech waveform recreated using peak sample points. Qualitatively, the reconstructed waveform is intelligible and sounds like a filtered version of the original waveform. This filtered sound corresponds with the shortcoming of the system, which are described in the next section.

samples for this 0.607 second speech clip. The highest frequency captured by the system was about 7.9kHz while the average was 917Hz. If a traditional system were to gather this much information, it would have to sample at 15.8kHz, resulting in massive 9,708 sample points. In other words, a traditional system would have to sense a signal, convert it to the digital domain, and then store it or process it 17.5 times more than a peak sampling based system.

Figure 3.5: A section of the speech waveform that was reconstructed in the previous figure. The Bézier reconstruction method was applied to the peak samples of the original waveform. The actual reconstruction was then resampled at 15.8kHz, as would be required by a traditional system for this particular waveform.

3.3 Reconciliation with Digitally Sampled Information

For the information sampled by the peak reconstruction method to be useful, typically it will have to be resampled at a rate consistent with a high-resolution digital component of the system. An example of a situation where this would be desirable is in the capture of a speech waveform. The digitally sampled portion of the event may not contain the entirety of the desired speech clip. This loss could be caused by a slow wake-up or even a slightly late wake-up event trigger. In any case, it may be useful to stitch the peak-sampled prelude information to the higher-resolution, digitally sampled information. In this way, the entire speech clip can be broadcast to the base station or processed further locally.

For a situation where a series of peak sampled points needs to be resampled, one only needs to calculate the times at which new samples are to be taken to utilize within the Bézier framework. An algorithmic method of doing so would be that for any time T , the Bézier reconstruction equation for time can be used in conjunction with the sample's spanning

Figure 3.6: An example of a change in the derivative between two peaks that does not generate extra local maximums or minimums. This information is forfeited by peak sampling.

peaks to calculate the x value. This time value can then be used in the amplitude equation, again with the appropriate spanning peaks, to calculate the amplitude for this sample. This process can then be repeated throughout the peak sampled data so that it can be resampled according to a digital system's requirements (Fig. 3.5). This resampling represents the local processing discussed before which can be used for machine-learning applications

3.4 Disadvantages of Peak Sampling

The first shortcoming of our method includes error which aggregates when a signal is not properly band-limited. In this case, it is possible for a higher-ordered harmonic to cause a noticeable change in the derivative of the signal between peaks (Fig. 3.6). This harmonic distortion creates slight plateau that does not quite generate any local maximums or minimums. Because it does not generate any peaks, it does not generate any change in the sampling rate. This high frequency component will be effectively filtered by the use of

peak sampling and the information contained will be irrecoverable.

Another shortcoming of this method is that the derivative of the signal between peaks is fixed and does not adapt to the signal. While this can be partially mitigated with some training data to identify the most effective fixed derivative for the target signal, we presently have no method for adapting this parameter on-the-fly.

3.5 Chapter Summary

This chapter has presented an irregular sampling method which adapts itself to the changing characteristics of a target signal. It was shown how the presented method yields a reduction in samples when compared to traditional Nyquist sampling. A reconstruction method which takes advantage of the irregular sampling method was also presented. Finally, special considerations and trade-offs germane to the practical implementation of this sampling and reconstruction method in a resource constrained system, such as a memory buffer, were presented. This sampling method will be implemented in a low-power buffer system in the following chapter.

Chapter 4

Analog Buffer Circuits

Wireless sensor networks (WSN) are one type of resource-constrained system that particularly benefit from the use of low-power sleep states. During a sleep state, a device halts normal operation to conserve energy. The device can then be signaled to leave the sleep state by a wake-up circuit so that an interesting event can be observed. The trade-off faced by designers of WSNs is maximizing the time a node spends in the sleep state to conserve power while minimizing the number of events that the node misses.

Many techniques exist to optimize the duty cycles of WSNs and ensure that interesting events are not missed by the sensor nodes [10, 36, 9, 11, 8]. However, all of these techniques entail the possibility of losing some interesting data just before the event occurs or during the moment between the detection of the event and the node fully waking up. For applications where the prelude to an event may contain valuable information, a memory buffer must be included to continuously record the data before an event is detected.

While a digital memory buffer could be implemented in a WSN node to record the input during a sleep state, its use would negate much of the power savings garnered by the use of sleep states. A digital memory buffer's storage process involves reading samples through an analog-to-digital converter and then storing the values in some type of digital memory. The sampling rate needs to be at least the Nyquist rate of the highest frequency component of the signal in order to preserve the information contained in the signal. Continuous sampling at such a high rate is costly in terms of both power consumption and real estate (i.e., area required for storing the sequence of digital values) [33]. These drawbacks make digital

Figure 4.1: To avoid the data loss normally associated with sleep states in WSNs, we introduce an ultra-low-power analog memory buffer to operate in parallel with a wake-up circuit. This system records data from the sensor while the WSN mote is allowed to remain in a sleep state, enabling signal approximation without sacrificing the energy savings from the sleep state.

memory buffers undesirable for applications in which ultra-low-power operation is a necessity.

In this chapter, we demonstrate an ultra-low-power analog memory system that asynchronously samples a signal and then stores those samples in a buffer. This system operates in parallel with a wake-up circuit (Fig. 4.1) to store signal information while the rest of the system sleeps. To further reduce the power consumption and size of the analog buffer, we utilize the peak sampling and reconstruction method described in detail in the preceding chapter.

4.1 System Overview

Traditional Nyquist sampling of a signal necessitates sampling at a frequency greater than twice the highest frequency component of the original signal. Using this constant rate of sampling leads to oversampling when the signal does not contain the highest allowable frequency components. Wasting system resources on superfluous samples is a primary source of inefficiency in memory buffers.

Figure 4.2: Our analog memory buffer locates the local maxima and minima in real time and stores their respective amplitudes and times separately.

A system designed for minimal power consumption should be able to adapt in real-time to the changing frequency of the signal being sampled to avoid resource inefficiency. In the previous chapter, we demonstrated how sampling only the local maxima and minima (both amplitude and time) provide sufficient information about a time varying signal to produce an adequate approximation. We will now build a working memory buffer upon the principle of that sampling method.

Max/min sampling is achieved through the use of a multi-stage analog system (Fig. 4.2). When a local max/min is detected, its voltage value and the time since the last max/min are stored in an analog memory buffer. These values can then be accessed later for reconstruction or for further analysis.

Within our analog memory system, the max/min locator generates a pulse when the signal needs to be sampled. This pulse enables the write phase in two different analog memory storage units, which are implemented as an array of sample-and-hold circuits. This write pulse causes the memory units to record the voltage value as well as the time since the previous max/min. This sampling scheme results in a series of voltage-time pairs that can be used to approximate the original signal. In the following sections, we discuss each of the circuits used to create these conditions.

Figure 4.3: A basic operational transconductance amplifier.

4.2 Operational Transconductance Amplifier

The operational transconductance amplifier (OTA) is one type of amplifier which is used heavily throughout this work. An OTA is essentially a voltage controlled current source, similar to an op-amp, which is a voltage controlled voltage source. In fact, an op-amp is basically an OTA with a voltage output stage. As such, it is sufficient to think of the OTA as a current producing op-amp.

$$I_{out} = G_m * (V_+ - V_-) \quad (4.1)$$

$$G_m = \frac{I_b}{2kT/q} \quad (4.2)$$

The core of the device is a differential pair which divides a bias current among its two branches as dictated by their differential voltage (Fig. 4.3). One branch is then mirrored to the other, creating an output node which either sinks or sources a current, which is again proportional to the differential voltage of the differential pair (Eq. 4.1 and Fig. 4.4). The device can be further manipulated by changing the transconductance, which is proportional to the bias current as seen in Eq. 4.2, where k is Boltzmann's constant, T is temperature, q is the charge of an electron, and the quantity $2kT/q$ is known as the thermal voltage.

Figure 4.4: A sweep of the output current versus the differential input voltage for a typical operational transconductance amplifier.

Figure 4.5: A more advanced operational transconductance amplifier.

One limitation of the OTA is that the the output current has some dependence on the voltage at the output node. The upper limit of the OTA is slightly less than the supply voltage, V_{dd} , which is required for M_4 to remain in saturation. The lower bound however is determined by the voltage on the drain of the bias transistor, V_x , and causes the lower

Figure 4.6: The symbol used to denote operational transconductance amplifiers throughout the rest of this work. The specific OTA referred to is the nine transistor version.

limit to be the difference of the minimum of the input voltages and the bias voltage [37]. To avoid this limitation, the more complex implementation of the OTA shown in (4.5) and denoted by the symbol (4.6) was used throughout this work. By mirroring both branches of the differential pair to the output node, the limitation on the lowest permissible output voltage is mitigated, making the new limit nearly ground.

4.3 Peak Detector

The max/min locator circuit is used to trigger the write phase in the short-term memory, so that only maxima and minima are sampled. Detection of local maxima and minima is achieved through the use of symmetric max/min locator circuits operating in parallel. Because of this symmetry, we will only discuss the maximum locator.

The core of the maximum locator circuit is the envelope detector (Fig. 4.7), which was introduced in [38]. A pair of transconductors compare V_{out} , the value stored on the capacitor by the envelope detector, with V_{in} , the system input. When $V_{in} > V_{out}$, the top path is active and the capacitor is charging; otherwise, the bottom path is active and the capacitor is discharging. By biasing $G_{m,A}$ such that $G_{m,A}/C_{PD}$ is at least on the order of the signal frequency, V_{out} will track V_{in} up to the local maximum. Then, by using a weaker biasing for $G_{m,D}$, V_{out} will begin to lag behind V_{in} as the signal decays.

The operation of the on-chip max/min locator can be seen in the top pane of Fig. 4.8. When V_{out} lags V_{in} , the comparators of the max/min locator circuits produce an output of logic high. On the rising edge of the output of either comparator, the pulse generator signals the memory system to enter the write phase and sample the input (Fig. 4.8).

Figure 4.7: The maximum locator circuit consists of an envelope detector, a comparator, and a pulse-generation circuit. The envelope detector tracks the input on the rising slope of the signal and then slowly decays on the falling slope of the input signal. This tempered decay rate causes the comparator to detect a discrepancy at the point when the signal begins to decay, i.e. the local maximum. The comparator then signals a pulse generator, which triggers the write phase in the memory storage unit. The minimum locator circuit operates similarly, tracking the input on the falling slope and lagging the input on the rising slope.

In biasing the adaptive decay structure, the architect faces a few inherent trade-offs. First, a slower decay will result in a situation where a small peak will be missed if it follows a larger peak. However, a slower decay benefits the comparator which requires a minimum difference between the input signal and the output signal, called the comparator resolution. The comparator resolution also manifests itself as a time-delay in detecting the peak values of the signal. Aggressive biasing of the comparator can lessen this, however it will likely result in more false positives.

For flexibility in defining the timing of the pulse generator and memory addressing, we implemented the pulse generator using an off-chip complex programmable logic device (CPLD). In a subsequent version of this system, the CPLD functionality will be integrated with the rest of the system.

Figure 4.8: The max/min locator circuits operate by tracking the input on the rising/falling slope, and then decaying at a slower rate. The discrepancy between the decay of the max/min locator circuit and the original signal is detected by the comparator, which in turn signals the pulse generator to activate a write pulse. These data were taken from a chip fabricated in a $0.5\mu\text{m}$ standard CMOS process available through MOSIS.

4.3.1 Track-and-Hold

A simplification of the envelope style that we are investigating is the detect-and-hold style of peak detector (Fig. 4.9). Again, the device is symmetric, therefore we will only

Figure 4.9: The track-and-hold style peak detector, with respect again to the maximum locator, consists of a rising edge tracker, a comparator, a pulse generation circuit, and a transmission gate controlled by the minimum locator's pulse generation circuit. Essentially, the device will track rising edge of the input signal in an attempt to detect local maximums and will reset to the input during the only time that one can be certain that a maximum has not occurred and is not presently occurring, during a minimum.

discuss the maximum locator portion of the device. This style is similar to the envelope style with the decay stage removed. In place of the decay stage is a switch that resets the output to the present input signal amplitude. This style has two modes of operation. The first mode is tracking mode. This mode is similar to the adaptive-decay structure, where the output of the circuit tracks the input signal on a rising edge. The second mode is hold mode. This mode results from a decline in the input signal which, due to the lack of a decay stage, results in the present output voltage being held. Like the adaptive decay style, it is this departure from the input signal which results in a peak location being detected. Unlike the adaptive decay style, the tracking mode is only resumed when the symmetrically opposite circuit, the minimum detector for our example, detects a peak. When this occurs, the switch is used to short the output node to the input node, thus resuming tracking mode.

This style of peak detector effectively mitigates the temporal resolution restrictions created by the slow-decay stage of the envelope style. Put simply, one does not have to be concerned about the decay stage being too slow to intercept a new peak. The resolution restrictions of the comparator, however, remain a concern. Also, this style is more sensitive to under-damping of the OTAs. If an OTA is under-damped, a ripple in the output signal could occur when it is reset from hold mode to tracking mode. This ripple about the input could cause a series of false peak detections which would be compounded by the fact that the

Figure 4.10: Simulated output of the track-and-hold style peak detector. The input wave is traced by a maximum locator circuit. The minimum pulse generator causes the maximum locator to reset to input. The minimum locator circuit behaves similarly.

opposite stage would then have to go through a series of premature resets, possibly creating a ripple of its own. This combination of false peaks and early resets could be recovered from, but it would create a great deal of wasted samples.

4.3.2 Crossing Style

An alternative to the envelope style peak detectors is crossing style peak detectors. In this style, an output wave is produced which pierces the input wave at local maximums and minimums (Fig. 4.11). This output wave amplitude is scaled according to the biasing and sizing of the circuit. The ability of the circuit to produce the desired output is frequency-limited. Below its frequency range, it will produce an output that mirrors the input. Above its frequency range the output slews too slowly, causing late detection of the peaks.

The unique operation of this circuit topology begins with one (Fig. 4.12) or two (Fig. 4.13) operational transconductance amplifiers (OTAs) biased in a follower configuration with the output stage connected to a capacitor. The output stage is then mirrored and connected to another capacitor of a smaller size. This smaller size creates a smaller time constant. Because the smaller capacitor is not in the follower branch, it is charged and discharged at a rate determined by the larger time constant for this secondary output stage. However,

Figure 4.11: Simulated output of a crossing style peak detector. The output wave crosses the input at only the local maximums and minimums. This sharp transition triggers the comparator to create an upward edge on the minimums and a falling edge on the maximums.

Figure 4.12: A crossing style peak detector constructed with a single OTA

Figure 4.13: A crossing style peak detector constructed with two OTAs

for a specified frequency range they will essentially have a virtual-short between them at local maximums and minimums. This virtual-short is caused by the input wave's derivative changing, causing a switch from charging to discharging, or vice versa, in the follower configuration. Unfortunately, this operation only occurs reliably across two or three specified decades of frequency.

We are currently exploring ways of mitigating the frequency restrictions of this unique topology by investigating alternative OTAs. One design that appears promising is based on the adaptive bias OTA developed in [39]. This OTA utilizes two differential pairs so that it can essentially shunt extra current to the output stage when the bias is insufficient. Our design (Fig. 4.14) takes advantage of this to create a greater range of operational frequencies. This extra frequency range however comes with a cost in precision. The time constants of the dual output stages are essentially skewed by the current shunting. This skew causes the detection of the peaks to generally be somewhat late.

Figure 4.14: A crossing style peak detector constructed with a single self-biasing OTA

4.4 Short-Term Memory

The short-term memory subsystem records the amplitude of the input wave at the time of max/min detection. The memory subsystem is formed by a 64-element array of sample-and-hold circuits that are accessed through a break-before-make multiplexer (Fig. 4.15). Read and write operation is controlled by the pulses received from the max/min locator circuit. The memory system continuously loops through the capacitors until a wake-up event has been detected and the samples need to be read.

The hold time of the storage capacitors is limited by thermal noise and the leakage through the reverse-biased p-n junctions in the transmission gate switches; this hold time can be specified by proper sizing of the capacitors at design time. However, increasing the size of the capacitors for increased precision will come at the cost of an increase in power

Figure 4.15: The capacitor bank tracks the system input and enters write operation only when the max/min locator’s pulse generator outputs a logic high signal. After each write, the address of the selected capacitor is incremented through the break-before-make multiplexer.

consumption and area of the memory system. The latter point is particularly important, as the size of this subsystem dominates the total area of the device.

Experimentally, we have determined that this analog implementation carries with it some challenges, including charge sharing and leakage, that may ultimately make it unfeasible to use in systems where even moderate precision is desired. For this reason, it is prudent to consider the cost of a digital implementation of this particular subsystem.

Human speech is considered a relatively high frequency event in the context of WSNs. Therefore, it will serve as a good standard to measure the viability of a digital memory subsystem. We will consider a system constructed of off-the-shelf parts including an analog to digital converter (ADC) (Part# AD7467) and a static random-access memory (SRAM) (Part# 23A640). In reality, the system would also consist of an impedance buffer and a state machine to control memory addressing, but for this calculation we will assume that the active current of the ADC and the SRAM dominate the power consumption.

The common supply voltage of the chosen ADC and SRAM is 1.8v, while the active current is $186\mu\text{A}$ and 6mA respectively. If we use the average sampling frequency from the voice application demonstrated in Chapter 3 of 917Hz , and assume it takes $4.7\mu\text{s}$ for a single sample to be converted and stored, then we are left with a power consumption of

Figure 4.16: The timer is a basic capacitive time-to-voltage converter. A bias current linearly discharges a capacitor. The charge on the capacitor can be captured, stored, and then easily processed into a time value utilizing the known rate of discharge. The capacitor is reset after every sample.

about $48\mu\text{W}$. While this power consumption would likely dominate any implementation of an analog memory system, it does not necessarily negate the energy savings from the use of sleep states. It is also worth noting that this power consumption could likely be greatly reduced by utilizing on-chip components designed with an intended specific application, as opposed to using off the shelf parts. In any case, it is possible that for certain high-precision applications, a digital implementation of this subsystem could be desirable.

4.5 Timing Circuit

The timing circuit utilized by this system is a capacitive time-to-voltage converter (Fig. 4.16). After each max/min is found, the capacitor is set to a known value and is then discharged by a constant current source. Then, when the next max/min is found, the capacitor's voltage is sampled and stored in the same manner as the max/min voltages. The stored voltage, along with the known charging rate for the capacitor, can be used to calculate the time between maxima and minima (Fig. 4.17).

Figure 4.17: Simulated output of the timer. The output waveform is the result of the linear discharge of the capacitor.

It is necessary to have some knowledge of the maximum and minimum possible time intervals which are likely to be measured prior to implementing this circuit. A knowledge of the minimum possible time will allow the architect to set the discharging rate aggressively enough so that the capacitor can be discharged by some non-negligible amount. Knowledge of the maximum possible time will prevent the architect from being so aggressive with the discharge rate that the capacitor discharges completely before the entire time period to be measured has passed. In practice, it is actually best that the discharge rate be set so that the capacitor does not discharge past a couple hundred millivolts, at which point the discharge rate is considerably less linear. Also, it is worth noting that for some application where the possible time periods vary greatly, a system with multiple capacitors set to different discharge rates could be employed. The downside of such a system would be a higher cost in terms of storage required to sample each of these extra capacitors.

4.6 System Examples

To demonstrate the range of operation of this analog memory buffer, several input signals were analyzed with a single set of bias points. While the system's biasing can be adapted to capture a large range of signals, this section will focus on the capability of the system for a single biasing condition.

These demonstration system tests were performed using the on-chip adaptive decay max/min locator. The multi-stage adaptive max/min locator was chosen because it performed over a wider variety of signal characteristics compared to the other fabricated peak detectors. The timing, pulse generation, and storage components were performed using MATLAB. The outputs of the on-chip max/min locator were obtained via a data acquisition board. While the storage and timing components were fabricated and verified through simulation and/or testing, we chose to use MATLAB to emulate them instead. Utilizing these devices would have required compensation for charge leakage and sharing. While these components are important, they primarily provide timing or holding of values; thus, using idealized MATLAB functions to represent them more clearly demonstrates the performance of the max/min locator, which is the core of the analog memory buffer and the associated algorithm.

The max/min locator was found to operate on signals with amplitudes ranging from 30mV_{pp} to 1.5V_{pp} and on signals with frequencies ranging from 30Hz to 30kHz. Figure 4.18 demonstrates finding max/min locations on an AM modulated waveform as well as on a chirp signal.

The system was also shown to reliably capture the local maxima and minima of a voice signal. Figure 4.19 shows that the majority of the wave was well approximated. Error found in the approximation of the original signal results from two sources. The first occurs from portions of the signal that are not properly bandlimited. We plan to correct this issue in future iterations by performing a filtering operation to remove high-order harmonics and high-frequency noise before attempting max/min location. The second source of error occurs when a small max (min) follows a large max (min) too closely in time. In this case, the max/min locator is unable to decay quickly enough to resume tracking the signal in

Figure 4.18: The system is capable of capturing a wide range of amplitudes and frequencies. For these plots, the input is shown in grey, detected max/mins are black dots, and the reconstructed wave is a black dashed line. These data were taken from a chip fabricated in a $0.5\mu\text{m}$ standard CMOS process available through MOSIS.

time for max/min detection. For these few portions of the signal, we are undersampling and cannot recreate the waveform exactly. However, we note that the speech signal that was reconstructed from the measured max/min values was intelligible, with low perceived

Figure 4.19: (a) The speech clip used as input to the system is shown in grey with detected max/mins in black dots. (b) The speech clip approximation from using the modified Bézier formula with the sampled max/mins. These data were taken from a chip fabricated in a $0.5\mu\text{m}$ standard CMOS process available through MOSIS.

distortion, even in the presence of these two types of sampling errors.

The reconstruction of the 1.68s speech sample demonstrated adequate approximation using a minimal amount of resources. In total, 652 samples were recorded, and the highest

frequency captured was 3125Hz. If a constant-sampling-rate technique were used, a sampling frequency of 6250Hz would be necessary, and a total of 10556 samples would have to be recorded. In addition to the 16-fold savings in sample storage afforded by the adapted sampling rate, the system had low power consumption: $1.17\mu\text{W}$ for the max/min locator and $0.586\mu\text{W}$ for each of the four buffers used in the storage system combine for a total power consumption of $3.52\mu\text{W}$. These power consumptions were simulated because we were unable to access individual components on chip.

In performing the reconstruction, we utilized the Bézier reconstruction method described in the previous chapter. We have found the Bézier equation to be a promising method for reconstructing a signal from its max/min values, as indicated by the reconstruction in Fig. 4.19.

4.7 Conclusions

This chapter has described a low-power system capable of recording a pre-defined duration of a signal, prior to the wake-up event of an embedded system. The results of the system were shown to be able to faithfully recreate a variety of waveforms. Operation across a range of amplitudes and frequencies was demonstrated, as was application upon a voice signal. The system, which was constructed in a $0.5\mu\text{m}$ standard CMOS process available through MOSIS, was found to consume only $3.52\mu\text{W}$.

In addition to the system presented, we have shown alternative designs for various parts of the system, particularly the peak detection unit. By addressing the shortcomings of each of these designs, we aim to develop a full on-chip system. The full system will contain the peak detector, timer, and memory as well as a state machine which stops sampling once an event has been detected and flushes the buffer samples. To make this system as accessible and practical as possible, it would be desirable for it to be built upon a reconfigurable and programmable architecture. The beginnings of such an architecture will be described in the following chapter.

Chapter 5

Reconfigurable Infrastructure for Analog Circuitry

As demonstrated in [40], analog signal processing (ASP) is a form of in-network pre-processing that can be used to process a signal locally, provide event detection for wake-up scenarios, and more. It is a compelling choice to use analog processing in wireless sensor networks (WSNs) due to the analog nature of real-world signals and due to the 20 year leap that analog electronics have been shown to have over digital systems in terms of performance-per-power consumed [14].

Analog sensor interfaces in WSNs, including the buffer presented in the previous chapter, tend to be application-specific and are not necessarily capable of operating in a range of other applications. Additionally, ASP design is lengthy, compared to digital systems which can make use of programmable and reconfigurable systems. Therefore, incorporation of an ASP front-end is typically more costly in terms of development costs.

For the architect of deployed WSNs, this means that highly specialized ASPs must be fabricated and purchased at a premium (relative to general-purpose reprogrammable digital platforms which benefit from economies of scale). This hindrance also means that ASP designs must be fabricated and tested individually, again slowing their progress relative to reconfigurable digital electronics.

To overcome these challenges, we recommend improving an ASP's implementation by making it reconfigurable. A reconfigurable architecture would allow a single ASP IC to be

Figure 5.1: A sensor node equipped with a field-programmable analog array (FPAA). The FPAA can be reconfigured in run-time to perform event detection and pre-processing at a power consumption that is significantly lower than the power that would be consumed by the mote’s built-in digital systems.

used for a variety of applications and to be updated in the field as its application is redefined. To that end, we have incorporated a field-programmable analog array (FPAA) within a WSN node in a manner similar to [41, 20]. In this chapter, we will describe its incorporation into a sensor node and further illustrate its use within a WSN with example applications.

5.1 FPAA Architecture

Previous FPAA designs have already demonstrated the ability to synthesize complex analog circuitry [18]; what we would like to establish is their viability in embedded systems such as WSNs. More specifically, we would like to establish an FPAA’s use within energy-constrained systems that monitor phenomena such as audio, vibration, and motion, which have sufficiently high bandwidths ($>100\text{Hz}$) to challenge the throughput of typical WSNs.

The FPAA we present here for use in WSNs consists of four computational analog blocks (CABs), including two for spectral analysis and two for subband processing (Figs. 5.1 and 5.2). The spectral-analysis stage consists of two bandpass filters in the form of capacitively-

Figure 5.2: Die photograph of the FPAA fabricated in a standard $0.5\mu\text{m}$ CMOS process available through MOSIS. The FPAA is 2.25mm^2 .

coupled current conveyors (C^4 filters), two envelope detectors, two adaptive-time-constant filters for suppressing envelope detector ripple, and two buffers. These subblocks all have tunable biases, allowing the user to perform several common types of analysis on a range of signals before further processing. Further processing takes place in the subband processing CABs, which provide the user access to elements of a smaller granularity, including operational transconductance amplifiers (OTAs), current mirrors, individual transistors, and capacitors.

A notable feature of this FPAA architecture is that, because it concentrates on relatively high-frequency phenomena such as certain types of simple harmonic motion, it provides both general analog computing tasks, and it is also suitable for parallelized processing. The latter feature stems from the fact that the subbands of the architecture can be configured to perform identically, and the architecture itself could easily be scaled up to further this task.

In our FPAA, reconfiguration is achieved via programmable switches in the connection box (which is used for intra-CAB routing) and the switch box (which is used for inter-CAB routing). The connection box consists of a full-crossbar configuration for flexible local routing. In the switch box, we implemented a variety of connection types (such as crossbar, crossover, and four-way switch points) to evaluate their value within sensor networks. To facilitate ease of integration with a sensor node, we implemented these switches using SRAM-controlled transmission gates. Each switch had an SRAM memory cell that set it to “on” or “off.” To load values, we used a row-by-row method that would load the state of all 16 switches in a given row. The configuration was written into the SRAM array using an on-chip serial peripheral interface (SPI). In total, the FPAA had 1436 switches, with the potential to route 40 unique nets.

5.2 Mote Interfacing

Our goal with this design was to integrate the FPAA into a WSN sensor node in a way that would enable us to easily monitor a range of phenomena. To that end, we created a printed circuit board (PCB), as shown in Figs. 5.3 and 5.4, that includes a variety of sensors, two FPAAs to enable scalability, a TelosB mote connector, a digital-to-analog converter (DAC)

Figure 5.3: PCB for interfacing the FPAA to a WSN sensor node. This board incorporates a variety of sensors, a CPLD, a DAC (on the underside of the PCB), and a socket for connecting a TelosB mote.

Figure 5.4: A high-level schematic of the PCB for interfacing sensors, the FPAA, and a WSN mote.

for providing bias voltages, as well as a complex programmable logic device (CPLD). For the sensors, we chose to focus on the relatively high-frequency phenomena that are traditionally very taxing on a WSNs power budget, including motion, audio signals, and various forms of simple harmonic motion. To monitor these phenomena, we equipped the board with a gyroscope, two microphones placed at opposite ends of the board to enable directional sensing, and a mini-stereo port to enable future expansion.

By including two FPAA's on the same board, we were able to effectively scale up our architecture, overcoming our IC space limitation, and build more sophisticated ASP designs. The FPAA's include SPI blocks that can be programmed directly through the attached TelosB mote's general purpose input/output (GPIO) pins. The on-board CPLD was used to minimize the number of TelosB pins that were used for digital I/O, thus freeing up more pins to be configured as ADCs. Additionally, the CPLD was used to define even more complex wake-up events. We also simplified setting individual bias points for all of the ASP blocks by including DACs which can be set and adjusted directly by the mote.

5.2.1 User Interface

We developed a software user interface to aid in the reconfiguration and tuning of the FPAA. This user interface simplifies the process of synthesizing the analog circuits on the FPAA to aid WSN designers who may not have the circuit-level expertise usually required to construct an ASP. The current implementation of the user interface allows users to construct individual ASP blocks and see the routing that the device will use to connect them. Once the user is satisfied with the design, the configuration, which consists of the switch settings and the DAC bias values, is converted to a header file by a Matlab script. The updated header file is then uploaded to the base-station mote running TinyOS, and could then be wirelessly transmitted to the remote nodes. The remote node then applies the new configuration to its FPAA.

Figure 5.5: (a) Example of a configuration of switches. (b) Implementation of how this configuration would be transmitted using our compression scheme.

5.2.2 Compression

When designing an FPAA for use in a WSN, the size of the FPAA is a critical design choice. While it is desirable to have an FPAA that is large enough to create sophisticated ASP designs, care must be taken to minimize the overhead of delivering and storing large configuration files within the network. The naïve approach for handling configuration files is to simply transmit the raw bits that will eventually be shifted into the FPAA. For example, the configuration file for the switch configuration in Fig. 5.5(a) will consist of 64 bits, only three of which are “on,” which implies that the configuration is redundant. If this method were scaled to large FPAAs that have 74,000 switches [18], for example, then significant energy would be wasted transmitting and receiving redundant bits.

To address this problem, we have developed a compression method that is inspired by entropy coding, but that is informed by our observations about typical FPAA configurations. Configuration files tend to be small. Therefore, traditional methods which utilize a codebook would have too much overhead (e.g. Huffman coding). Even when considering FPAAs of larger scale, ASP algorithms tend to have a parallel nature and are still amenable

Figure 5.6: Top-level schematic of the circuit synthesized in the FPAA to demonstrate its spectral analysis capabilities. The circuit detects portions of the signal where the frequency content rises in the 2-4kHz range. The “Correlation” stage detects the simultaneous presence of content in the high-frequency band and the delayed low-frequency band. The “Inhibition” stage nulls the output when content is present in the low-frequency band to avoid triggering on wideband signals. G_{m2} is biased by the gate of the attached pFET while the output current of G_{m1} is mirrored through the diode connected FET. Also note that $V_{pulldown}$ is a constant bias used to weakly pull down the output when G_{m2} is shutoff by G_{m1} .

to compression. An example of this would be a large filter bank which utilizes the same operation in each sub-band. This identical operation means that the switch settings would be the same in all channels; therefore, it would only be necessary to transmit the settings for one channel and then apply those settings to the remaining channels.

Due to redundancies in the switching matrix, most rows tend to have no switch set. Therefore, we begin our row-by-row configuration scheme by delineating whether or not any switches are set within a given row. Only if a switch is set in the row do we specify the location of the switch within the row using a four bit identification number. An example compression is shown in Fig. 5.5(b), where the 64-bit configuration of Fig. 5.5(a) is successfully reduced to 19-bits. The size of the compressed configuration depends upon the number of “on” switches N_{on} , and is equal to $5N_{on} + N_{rows}$, where N_{rows} is the number of rows in the FPAA. We have determined experimentally that the energy for the mote to decode the configuration is $34.1\mu\text{J}$, while the reduction in transmitted data saves 3.5mJ .

Figure 5.7: Spectral analysis performed by the analog IC. (Top, Middle) The transient plot and the spectrogram plot of the input signal, respectively. The sinusoidal signal, which includes Gaussian noise, varies from 1kHz to 8kHz and concludes with a steady input of ten sine waves ranging from 2kHz to 4kHz. (Bottom) The output stage successfully detects portions of the signal where the frequency content rises in the 2-4kHz range.

5.3 System Examples

To illustrate the functionality of FPAA-based ASP designs in WSNs, we connected our FPAA interface board to a TelosB mote and synthesized several signal-processing circuits on the FPAA. Each of these circuits could be used to generate wake-up signals to turn on the TelosB mote. In each scenario, all reconfiguration commands were sent over the radio through another TelosB mote, which acted as the base station.

The first system that we demonstrate shows ability of the FPAA to perform basic spectral analysis (Figs. 5.6 and 5.7). Here, the FPAA has been configured to analyze a signal's frequency content and detect a rising frequency in the 2-4kHz range. The signal is first filtered through parallel bandpass filters set to center frequencies of 2kHz and 4kHz. The lower-frequency signal (x_0) is then delayed. A cascade of OTAs computes the product of the delayed low-frequency signal with the instantaneous high-frequency signal (x_1), thus

Figure 5.8: Schematic of the implemented voice-activity detection algorithm. The device triggers an event when the amplitude modulation in the speech band occurs at a rate that is typical of speech.

providing a measure of simultaneity, reminiscent of the motion-analysis system in [42]. To ensure that static wideband signals do not trigger the detector, the final portion of the circuit pulls the output low when x_0 is high. The resulting output is high only when the x_1 and the delayed version of x_0 are high and the x_0 is not high. As a result, a pulse is generated when the signal is rising in the correct frequency range.

The next system demonstrates the ability of the FPAA to implement a voice-activity detector, based upon the scheme presented in [43] (Figs. 5.8 and 5.9). Audio signals were first passed through the spectral analysis CAB where they were filtered from 10 Hz to 2kHz using a bandpass filter. The envelope of this speech band was then found and passed through another bandpass filter, with corner frequencies at 2Hz and 12Hz corresponding to the phoneme band. The magnitude of the phoneme band was then used to trigger a time-to-voltage converter that would create a ramping voltage when the phoneme band exceeded a specified threshold. The time-to-voltage converter then triggers an event when this ramped voltage exceeded a threshold. While the inclusion of the time-to-voltage converter and the subsequent comparator stage may seem redundant at first, it allowed the device to operate in noisy, non-idealized conditions. The signal at each stage is shown in Fig. 5.8, and it is shown that the speech portion of the input signal was correctly identified in the presence of noise. The overall output can be used to identify to the rest of the sensor node that a signal of interest has been found.

Figure 5.9: Demonstration of the analysis of a signal throughout the voice-activity detection algorithm. For this test, the input was a male voice corrupted by noise from an airport environment, at a signal-to-noise ratio of 10dB.

The final synthesized system partially implements the buffer system introduced in the previous chapter (5.10). This application pairs the gyroscope with two of the adaptive-decay peak detectors previously introduced. When a peak detector generates a pulse, signaling the occurrence of a local maximum or minimum, an interrupt pin is triggered on the Telos mote. The interrupt causes the signal to be sampled and stored. When the allotted memory is full, the samples are streamed back to the base station where they can undergo reconstruction. By using the reconfigurable architecture to implement the memory buffer, an architect could easily adjust the circuit implementation to better fit the data received by the gyroscope, or any other sensor type.

Figure 5.10: A block diagram showing the implementation of the peak detection algorithm in conjunction with a gyroscope.

5.4 Conclusions

In this chapter, we have presented our FPAA architecture which was designed for use in WSN systems. Also, we have shown how an FPAA can be easily integrated into a WSN system, including those which monitor phenomena of relatively high-frequency content. Finally, we demonstrated the system's ability to recreate complex ASP designs.

Chapter 6

Conclusion

In this work, I have attempted to advance the use of analog signal processing (ASP) as a means to extend the lifetime and viability of energy constrained sensing systems. I have presented peak sampling and reconstruction, which is a method capable of adjusting its sampling rate to a signal's changing frequency content while still maintaining the ability to produce a faithful recreation of the signal. A buffer was presented that takes full advantage of peak sampling - consuming a minimal $3.52 \mu\text{W}$ for the presented application. And finally, the infrastructure for implementing such a system into a wireless sensor network (WSN) was considered. This infrastructure was reconfigurable and reprogrammable, thus it not only addressed issues inherent WSN deployment, but it also widened the range of people who can design ASP front-ends past those with circuit-level expertise. These works have resulted in two conference publications, including a lecture and a poster presentation, as well as a refereed demonstration at a conference and a full utility patent application. Despite all of this, there is still a good deal of work to be done with this research.

There is still considerable leg-work to be done with proving the viability of peak sampling. A thorough analysis of this would include not only placing theoretical bounds on the method's capabilities, but also finding practically what types of signals benefit the most. In addition to this, Bézier reconstruction should be further explored. It would be worthwhile to put error bounds on this method of reconstruction and also to begin work on finding a method to tune its concavity parameters on-the-fly in a machine-learning environment. Beyond this, other methods of reconstruction that can take advantage of a base station's higher computing

potential and lesser constraints should be explored.

The analog memory buffer itself still has some way to go before it is a truly finished product. Specifically, I would like to consolidate the entire system onto a chip; this includes the peak locator, timer, memory system, as well as any circuitry required to begin and end sampling in response to a node entering or leaving a sleep state, respectively. More than this though, it is important that this system not be built as a stand-alone system. It should be configurable within the infrastructure of a field-programmable analog array (FPAA) to enable the maximum practical use within WSNs.

In addition to the memory buffer, I believe there are other types of systems that could benefit from the use of peak sampling and reconstruction. One particular system is analog-to-digital converters (ADCs). While a lot of my work has been done to mitigate their use, they are inevitably an important part of any system that attempts to interact with real-world analog signals. For this reason, it would be very worthwhile to see if an ADC built upon the premise of peak sampling could have a relatively low energy cost. This ADC could then be incorporated within larger systems, such as the analog buffer or the FPAA.

In future iterations of our FPAA design, we plan to scale up the size. This scaling will allow us to test and implement even more sophisticated ASP functionality. Also, we would like to implement the use of floating gate transistors to provide tunable bias currents. Providing on chip tunable bias currents would replace the use of digital-to-analog converters (DACs), both simplifying the testing platform and procedure as well as making the entire design more realistic for deployed networks. We also plan to continue our testing on ASP inclusion in WSNs, particularly with regards to adaptive ASPs. By utilizing FPAAs, we will be able to easily implement this adaptation and quantify its value added for varying application spaces.

Another improvement we hope to make is to the user interface system of the FPAA. The goal would be to have users provide a high-level description of the blocks they want to implement, and then have the program auto-route the optimized design. This would allow users with little circuit-level knowledge utilize the device and thus ASPs in their design of WSNs.

These improvements and future research directions are worthwhile in their own right.

They will answer questions and provide technologies that will greatly enhance our understanding of energy constrained embedded systems. For this reason, I plan to incorporate these research goals as part of my future work as I earn a Ph.D. at West Virginia University.

References

- [1] M. F. Duarte and Y. H. Hu, “Vehicle classification in distributed sensor networks,” *Journal of Parallel and Distributed Computing*, 2004.
- [2] B. Wei, M. Yang, R. Rana, C. Chou, and W. Hu, “Distributed sparse approximation for frog sound classification,” in *Information Processing in Sensor Networks*, Apr. 2012, pp. 105–016.
- [3] V. Raghunathan, C. Schurgers, and M. B. Srivatsava, “Energy-aware wireless microsensor networks,” *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 40–50, Mar. 2002.
- [4] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *International Conference on System Sciences*, 2000.
- [5] S. Muruganathan, D. Ma, R. Bhasin, and A. Fapojuwo, “A centralized energy-efficient routing protocol for wireless sensor networks,” *IEEE Communications Magazine*, vol. 43, no. 3, pp. 8–13, 2005.
- [6] J. W. Lee, J. Y. Lee, and J. J. Lee, “Jenga-inspired optimization algorithm for energy-efficient coverage of unstructured WSNs,” *IEEE Wireless Communications Letters*, vol. 2, no. 1, pp. 34–37, 2013.
- [7] R. R. Rout and S. K. Ghosh, “Enhancement of lifetime using duty cycle and network coding in wireless sensor networks,” *IEEE Transactions on Wireless Communications*, vol. 12, no. 2, pp. 656–667, 2013.
- [8] P. Dutta, M. Grimmer, A. Arora, S. Bibyk, and D. Culler, “Design of a wireless sensor network platform for detecting rare, random, and ephemeral events,” in *Information Processing in Sensor Networks*, 2005.
- [9] S. Jevtic, M. Kotowsky, R. Dick, P. Dinda, and C. Dowding, “Lucid dreaming: Reliable analog event detection for energy-constrained applications,” in *Information Processing in Sensor Networks*, 2007, pp. 350–359.
- [10] M. Malinowski, M. Moskwa, M. Feldmeier, M. Laibowitz, and J. Paradiso, “Cargonet: A low-cost micropower sensor node exploiting quasi-passive wakeup for adaptive asynchronous monitoring of exceptional events,” in *ACM Sensys*, 2007.

- [11] D. Goldberg, A. Andreou, P. Julián, P. Pouliquen, L. Riddle, and R. Rosasco, “A wake-up detector for an acoustic surveillance sensor network: Algorithm and VLSI implementation,” in *Information Processing in Sensor Networks*, 2007, pp. 134–141.
- [12] R. Sarpeshkar, “Analog versus digital: Extrapolating from electronics to neurobiology,” *Neuroal Computation*, vol. 10, pp. 1601–1608, 1998.
- [13] P. Hasler, P. Smith, D. Graham, R. Ellis, and D. Anderson, “Analog floating-gate, on-chip auditory sensing system interfaces,” *IEEE Sensors Journal*, vol. 5, no. 5, pp. 1027–1034, 2005.
- [14] S. Ravindran, P. Smith, D. W. Graham, V. Duangudom, D. Anderson, and P. Hasler, “Towards low-power on-chip auditory processing,” *EURASIP Journal on Applied Signal Processing*, vol. 2005, no. 3, pp. 1082–1092, Jan. 2005.
- [15] B. Rumberg, D. W. Graham, and V. Kulathumani, “Hibernets: Energy-efficient sensor networks using analog signal processing,” in *Information Processing in Sensor Networks*, 2010.
- [16] R. Sarpeshkar, C. Salthouse, J. J. Sit, M. Baker, S. Zhak, T. K. T. Lu, L. Turicchia, and S. Balster, “An ultra-low-power programmable analog bionic ear processor,” *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 4, pp. 711–727, 2005.
- [17] R. R. Harrison and C. Charles, “A low-power low-noise CMOS amplifier for neural recording applications,” *IEEE Journal of Solid-State Circuits*, vol. 38, no. 6, pp. 958–965, 2003.
- [18] A. Basu, S. Brink, C. Schlottmann, S. Ramakrishnan, C. Petre, S. Koziol, F. Baskaya, C. M. Twigg, and P. Hasler, “A floating-gate-based field-programmable analog array,” *IEEE Journal of Solid-State Circuits*, vol. 45, no. 9, pp. 1781–1794, Sept. 2010.
- [19] E. K. F. Lee and P. G. Gulak, “A CMOS field-programmable analog array,” *IEEE Journal of Solid-State Circuits*, vol. 26, no. 12, pp. 1860–1867, Dec. 1991.
- [20] E. Mackensen and C. Muller, “Implementation of reconfigurable micro-sensor interfaces utilizing FPAA,” in *IEEE Sensors*, Nov. 2005, pp. 1064–1067.
- [21] C. Shannon, “Communication in the presence of noise,” *Proceedings of the IRE*, pp. 10–21, 1949.
- [22] N. T. Gaarder, “A note on multidimensional sampling theorem,” *Proceedings of the IEEE*, vol. 60, pp. 247–248, Feb. 1972.
- [23] L. L. Campbell, “A comparison of the sampling theorems of Kramer and Whittaker,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 12, no. 1, pp. 117–130, Mar. 1964.
- [24] J. Yen, “On nonuniform sampling of bandwidth-limited signals,” *IRE Transactions on Circuit Theory*, vol. 3, no. 4, pp. 251–257, 1956.

- [25] A. J. Jerri, “The Shannon sampling theorem - its various extensions and applications: A tutorial review,” *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1565–1596, 1977.
- [26] F. Bond and C. Cahn, “On the sampling the zeros of bandwidth limited signals,” *IRE Transactions on Information Theory*, vol. 27, pp. 110–113, 1958.
- [27] N. Sayiner, H. V. Sorensen, and T. R. Viswanathan, “A level-crossing sampling scheme for A/D conversion,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 43, no. 4, pp. 335–339, 1996.
- [28] I. Bar-David, “An implicit sampling theorem for bounded bandlimited functions,” *Information and Control*, vol. 24, no. 1, pp. 36–44, Jan. 1974.
- [29] S. Feizi, V. K. Goyal, and M. Medard, “Time-stampless adaptive nonuniform sampling for stochastic signals,” in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, 2012, pp. 3809–3812.
- [30] D. L. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
- [31] Z. Charbiwala, P. Martin, and M. B. Srivastava, “Capmux: A scalable analog front end for low power compressed sensing,” in *International Green Computing Conference*, 2012, pp. 1–10.
- [32] K. Kanoun, H. Mamaghanian, N. Khaled, and D. Atienza, “A real-time compressed sensing-based personal electrocardiogram monitoring system,” in *Design Automation Test in Europe*, 2011, pp. 1–6.
- [33] D. Wei and A. Oppenheim, “Sampling based on local bandwidth,” in *Asilomar Conference on Signals, Systems, and Computers*, Nov. 2007, pp. 1103–1107.
- [34] E. C. Titchmarsh, “The zeros of certain integral functions,” in *Proceedings of the London Mathematical Society*, vol. 25, January 1926, pp. 283–302.
- [35] G. Farin, *Curves and Surfaces for CAGD: A Practical Guide*. Academic Press, 1996.
- [36] B. Rumberg, D. Graham, V. Kulathumani, and R. Fernandez, “Hibernets: Energy-efficient sensor networks using analog signal processing,” *IEEE Journal of Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 3, pp. 321–334, Sept. 2011.
- [37] C. A. Mead, *Analog VLSI and Neural Systems*. Addison-Wesley, 1989.
- [38] B. Rumberg and D. Graham, “A low-power magnitude detector for analysis of transient-rich signals,” *IEEE Journal of Solid-State Circuits*, vol. 47, no. 3, pp. 676–685, March 2012.
- [39] M. Degrauwe, J. Rijmenants, E. Vittoz, and H. De Man, “Adaptive biasing CMOS amplifiers,” *IEEE Journal of Solid-State Circuits*, vol. 17, no. 3, pp. 522–528, 1982.

- [40] B. Rumberg, D. W. Graham, V. Kulathumani, and R. Fernandez, “Hibernets: Energy-efficient sensor networks using analog signal processing,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 3, pp. 321–334, 2011.
- [41] Y. Chen, C. M. Twigg, O. A. Sadik, and S. Tong, “A self-powered adaptive wireless sensor network for wastewater treatment plants,” in *IEEE International Conference on Pervasive Computing and Communications Workshops*, March 2011, pp. 356–359.
- [42] R. R. Harrison and C. Koch, “A robust analog VLSI Reichardt motion sensor,” *Analog Integrated Circuits and Signal Processing*, vol. 24, pp. 213–229, 2000.
- [43] T. Delbrück, T. Koch, R. Berner, and H. Hermansky, “Fully integrated 500 μ w speech detection wake-up circuit,” in *International Symposium on Circuits and Systems*, 2010, pp. 2015–2018.