2017

Data Conversion Within Energy Constrained Environments

Brandon M. Kelly

Data Conversion Within Energy Constrained Environments

Brandon M. Kelly

Dissertation 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

Doctor of Philosophy
in
Electrical Engineering

David W. Graham, Ph.D., Chair
Matthew Valenti, Ph.D.
Natalia A. Schmid, Ph.D.
Vinod Kulathumani, Ph.D.
Edward M. Sabolsky, Ph.D.

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2017

Keywords: Analog-to-Digital Conversion, Extrema Sampling, Floating-Gate,
Reconfigurable Analog, Wearable Devices, Wireless Sensor Networks

Abstract

Data Conversion Within Energy Constrained Environments

Brandon M. Kelly

Within scientific research, engineering, and consumer electronics, there is a multitude of new discrete sensor-interfaced devices. Maintaining high accuracy in signal quantization while staying within the strict power-budget of these devices is a very challenging problem. Traditional paths to solving this problem include researching more energy-efficient digital topologies as well as digital scaling.

This work offers an alternative path to lower-energy expenditure in the quantization stage – content-dependent sampling of a signal. Instead of sampling at a constant rate, this work explores techniques which allow sampling based upon features of the signal itself through the use of application-dependent analog processing. This work presents an asynchronous sampling paradigm, based off the use of floating-gate-enabled analog circuitry. The basis of this work is developed through the mathematical models necessary for asynchronous sampling, as well the SPICE-compatible models necessary for simulating floating-gate enabled analog circuitry. These base techniques and circuitry are then extended to systems and applications utilizing novel analog-to-digital converter topologies capable of leveraging the non-constant sampling rates for significant sample and power savings.

Dedication

Marriage is a promise, a potential,
made in the hearts of two people who love,
which takes a lifetime to fulfill.
- Edmund O'Neil

Dedicated to Erica and the lifetime we will spend fulfilling that promise.

Acknowledgments

First, thank you to my family for all of the love and support over the years. I'm lucky to have such a large family, and even luckier to feel so close with all of them.

Thank you to Dr. Matthew Valenti, Dr. Natalia Schmid, Dr. Vinod Kulathumani, and Dr. Edward Sabolsky for serving on my committee and thank you Dr. David Graham for leading that committee. I have been lucky enough to have learned from most of you for the past decade. Also, thank you West Virginia University and the WVU Foundation for supporting my research.

I would also like to thank Brandon Rumberg, Kyle Allard, Spencer Clites, Steven Andryzcik, Alex Dilello, Mir Mohammad Navidi, and Haifa Abulaiha. Working on a deadline through the night, through the weekend, or through a blizzard has a funny way of building a strong camaraderie. Working alongside you is what I will miss most about my time at WVU.

A very sincere thank you to anyone who has ever posted a free tutorial or lesson online on anything electronics related. Few get rich by sharing their knowledge online, but many like myself are better engineers for it.

And finally, to my wife Erica. You challenge me, you push me, and you help me discover what I am capable of accomplishing. Thank you.

Contents

List of Figures

Chapter 1

Introduction

It is now common to expect digital devices to be able to interact with our analog environment, but this interaction is really the melding of two different worlds. Watches that monitor our heartbeat, phones that listen for specific commands, roads that monitor themselves for degradation – these are all examples of digital systems interacting with our analog world. Converting analog signals into digital representations is very energy intensive, but it is a necessary prerequisite to our devices performing some useful computation or monitoring task. Despite the complexity involved in this conversion, these sensor interfaced-devices are proliferating at an incredible rate.

Some of these devices take the moniker of wireless sensor networks (WSNs) and are primarily focused on remote monitoring applications. These applications range from monitoring the integrity of roadways and bridges [1] to monitoring cargo in transit [2]. WSNs, as the name implies, form their own network over which to share information and help operators make informed decisions about applications which would usually be to expansive or remote for an individual to monitor. As these networks grow, the primary limitations become the individual power budgets of the individual nodes as well as the growing aggregate of data which must be passed by the entire network.

Another class of discrete-sensor-interfaced devices are bio-wearable technologies. Initially, these applications focused on health and exercise related goals (e.g., monitoring heart-rate during a jog or observing blood pressure over the course of a day), but have been rapidly extended to more clinically-driven applications within physical therapy and sports sciences

[3, 4, 5]. These devices collect incredible amounts and varied types of data, pushing our understanding of our own physiology. As the desire and need for more advanced bio-signal monitoring applications develop, so too does the need for devices which can be adapted to a variety of different processing algorithms, all without exceeding their limited power and data storage budgets.

The most broad class of sensor-interfaced devices are known collectively as the internet of things (IoT). IoT devices number in the tens of billions, and are only expected to keep proliferating. They range from smart-refrigerators and smart-thermostats to more energy constrained applications such as trackers and cell phones. The sheer number of applications and the volume of data processed in the IoT is staggering, and presents a challenge to engineers attempting to make use of it.

For energy-constrained sensor-interfaced devices to continue to proliferate and advance in complexity, new low-power architectures and signal-compressing methods must be developed. This chapter will highlight the traditional paths that have been taken to achieve both energy and sample reduction, while developing the background necessary to understanding the need for the devices and techniques presented by this work.

1.1 Shannon-Nyquist Sampling

An analog signal is defined as being continuous in time. Being continuous in time means that, between a start and end time, the signal is defined for any point in time, no matter how fine the time scale. These continuous-time analog signals are different than the discrete-time (i.e., defined only at specific time intervals) digital signals which empower the operation of modern electronics. The challenge in interfacing modern electronics with the real-world is in quantizing analog signals in a way that creates high-quality digital representations with low-power expenditures. This section will discuss the way in which signals are typically sampled and quantized.

One of the influential works that has made the digital age possible is Claude Shannon's "Communication in the Presence of Noise" [6]. In this work, Shannon created a foundation for how we quantize and transmit analog data. The rate, known as the Nyquist rate, was described succinctly by the following theorem:

> *If a function f(t) contains no frequencies higher than W cps, it is completely determined by giving its ordinates at a series of points spaced $\frac{1}{2W}$ seconds apart.*

This theorem states that a band-limited signal may be completely reconstructed if sampled at twice its highest frequency component. The band-limiting condition is a necessary prior step to choosing the Nyquist Rate. Intuitively this condition makes sense – it would be difficult to choose a rate based upon the highest frequency content of a signal if you did not limit what that frequency could be. The process yields the following equation:

$$f(x) = \sum_{k \varepsilon Z} f(kT) sinc(x/T - k) \tag{1.1}$$

As a sampling scheme, Shannon's method is theoretically sound and surprisingly simple to understand. However, the physical implementation is marred by some practical difficulty. Firstly, real-world signals are not bandlimited. Generally, this is both a natural result (feint harmonics in speech for example [7]) as well as an artificial product of high-frequency noise being added to the system. In a similar vein, perfect low-pass filters do not exist. Furthermore, the more stringent your filtering requirements, the longer the required computation – at least for digitally produced filters.

To handle these non-idealities, modern mathematicians have adjusted the definition so that a 'modern' approach to Shannon sampling is one that minimizes reconstruction error, given not-completely-bandlimited functions and non-ideal filters [8]. This reinterpretation has been realized in the form of different approximations, allowing quicker convergence with understood resolution trade-offs through the use of specific mathematical modalities including splines [9] and wavelet theory [10].

These mathematical insights have led to more efficient implementations of the Shannon-sampling paradigm, as well as a more thorough understandings of its limits, but the Nyquist rate remains a limiting factor. Simply put, the sampling and reconstruction algorithms can become increasingly refined, but no matter what, you will still have to sample the signal at twice its highest frequency. The Nyquist rate is a hard minimum on the number of samples required to perfectly represent a signal – at least for traditional Shannon sampling.

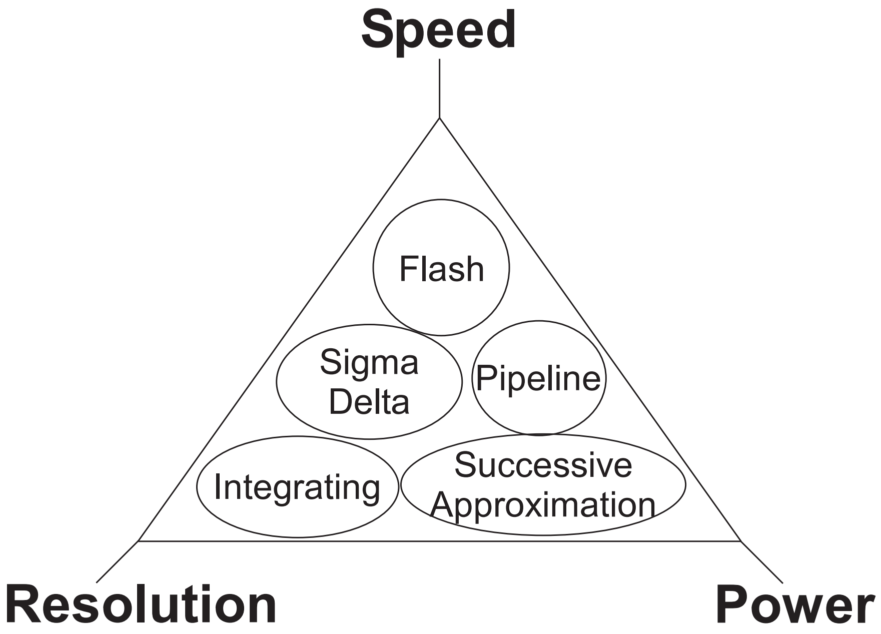


Figure 1.1: There is an intrinsic three-way trade between speed, resolution, and power within ADC design. On the continuum of this trade-off are the five primary ADC types. Speed refers to how quickly an ADC can perform a conversion. Power refers to the power efficiency of a particular topology. Resolution refers to how accurately, or with how many digital bits, an ADC can quantize an analog quantity.

1.1.1 Nyquist-Rate Data Converters

There are many topologies within the area of analog-to-digital converters (ADC), but they all tend to trade three quantities: power expenditure, conversion speed, and sample resolution (Fig. 1.1). Typically, advances in one quantity result in some loss in one or both of the other two. While there are examples within individual topologies which stand as outliers, examining the trends within these topologies [11] allows us to make some broad inferences about their optimum application.

The first category of ADCs is comprised of flash and pipeline ADCs. Both are known for their speed, but at the cost of very high power consumption. While this high consumption can be partially mitigate through the use of multiple lower resolution quantizers, with subranging or folding techniques [12, 13], they are still best left to applications which require giga-hertz rate sampling at the cost of milli-watt power consumption [11, 14].

On the other end of the Nyquist-rate ADC spectrum are very high-resolution ADCs –

particularly integrating converters. This topology can reach resolutions up to 20 bits, but is generally limited to sub-kilohertz sampling rates. This technique suffers from linearity and scaling issues and is no longer a popular area in research outside of very application specific circumstances [15].

Somewhere between these two extremes, there is a speed-resolution-power trade-off that is considered ideal for low-power sensor-enabled systems. This region is characterized by the successive-approximation ADC (SA-ADC) topology, which is easily implemented in energy-constrained systems.

SA-ADCs follow a conversion process that is very similar to a binary search. In essence, the SA-ADC has a list of ordered numbers that represent all of the digital values which it is able to estimate an analog signal to be. The SA-ADC finds where in this list the analog sample belongs by first comparing it to the midpoint of its list. If, for example, the analog value is found to be greater than this midpoint, the bottom half of the list is discarded. The top half of the list is then split in half again, and the process repeats itself until the closest approximation within the list to that sample is found. This approximation is then the digital word produced by the SA-ADC.

SA-ADCs are the chosen topology for most cutting-edge ADC research. Many interesting variants of SA-ADCs have been developed, including subranging [16], time-interleaved [17], variable resolution [18], and self-clocking architectures [19]. In addition to demonstrating some of the highest conversion rates among papers submitted to top-tier conferences [11], SA-ADCs have been shown empirically to be the most energy efficient topology. [20] showed that among the many types of ADC architectures, the efficiency of switch capacitor circuits in CMOS technology make SA-ADCs among the most inherently efficient for the low to medium resolution range.

Even though ADCs are not purely digital systems, they have still benefited from both process scaling and from a plethora of research. In [21], Murmann predicted that power expenditure in ADCs would continue to halve every two years, as long as CMOS processes continued to scale in the nano-meter range. And while the area of ADC research has made great strides in reducing power consumption, the ADC itself only represents one part of the energy expended by a discrete sensor-interfaced device.

1.2 Post Quantization Energy Expenditure

Within networks of discrete sensor-interfaced devices, there are many forms of power consumption. This work thus far has only discussed the cost of sampling and quantizing analog signals. For these quantized signals to be useful, however, they must normally be processed by a microcontroller and/or transmitted to some other device by a transceiver.

Microcontrollers are digital devices which process and route digital data. They are the necessary step, in traditional systems, for turning digital data into usable pieces of information. Unfortunately, this digital processing comes at a cost. One of the lowest-power microcontroller series on the market, the MSP430, runs on a supply voltage of 1.8 volts and an active current of 101.25μA/MHz. This level of current draw may be low for a microcontroller, but it is too high for an energy constrained system to maintain. It is thus important that as little active processing as possible be performed by the microcontroller so that it can remain in a lower power sleep-state, where it only consumes 0.5μA of current.

Transceivers transmit or receive data in wireless applications. They are necessary for communicating raw signals or processed data from WSN nodes, wearable devices, or IoT devices to more computationally powerful base stations. One example use-case is a wearable pedometer transmitting gait information back to a cell-phone for easier viewing and storage of information. A different use-case might be an IoT device transmitting a speech signal to a more powerful base-station for speaker and command identification. In either case, the number of samples in the signal is related to how long the transceiver(s) must be left in a run-mode. An example of a very low power transceiver is the CC110L series, which operates on a 1.8 volt supply with a 16mA active current draw and a 200nA sleep-state current draw. Clearly, it greatly benefits the power-budget to make the transmitted message as short as possible, thus utilizing the active-state of the transceiver as little as possible.

These components, transceivers and microcontrollers, represent system overhead in discrete sensor-interfaced devices. Reduction of their power consumption can take two forms. The first is simply through improvements in device topology or scaling of process. This form is the most direct method, but requires considerable research and, for process-scaling, increasingly costly fabrication processes. The second way to reduce power consumption is to reduce the required amount of data points which need to be processed and transmitted. If this reduction of data is not done digitally, then the next logical place for it to occur is within the analog domain, prior-to or during signal quantization. Whatever power is consumed during this data reduction must then be weighed against the power saved by reduced processing and transmission to determine the overall system impact.

1.3 Analog Signal Conditioning and Pre-Processing

Prior to the quantization of a signal, it is possible to perform some conditioning and classification operations. Since the signal is analog to begin with, analog electronics offer a natural and power efficient means of doing so [22]. In [23], for example, an analog auditory sensory system was presented 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.

Given the pre-requisite of band-limitedness for Shannon-Nyquist sampling, it is no surprise that most data converters include an analog signal-conditioning front end. In this context, signal conditioning typically includes performing some sort of filtering operation to reduce the presence of high-frequency noise within a signal. Intuitively, removal of this noise ensures a more accurate sampling and reconstruction of the target signal.

Another, less common operation is to use analog electronics to perform a classification operation. By taking a signal and breaking it down by frequency spectrum, magnitude, or periodicity, analog electronics are capable of detecting events of interest. These events could include the onset of voice [24, 25], the passing of a certain class of automobile [1], or the onset of a cardiac event within an electrocardiogram [26]. The detection of these events allows the system to selectively turn the quantization stage on only during events of interest.

Unfortunately, analog design does not benefit from process scaling the way that digital design does. In fact, scaling tends to result increased mismatch, noise, and non-linearity in analog circuits. A large part of this is caused by the lower supply voltage, which creates less ‘head room’ within which the analog circuit can operate. In addition to this problem, analog circuits are often very application-specific, requiring new circuits to be designed on a

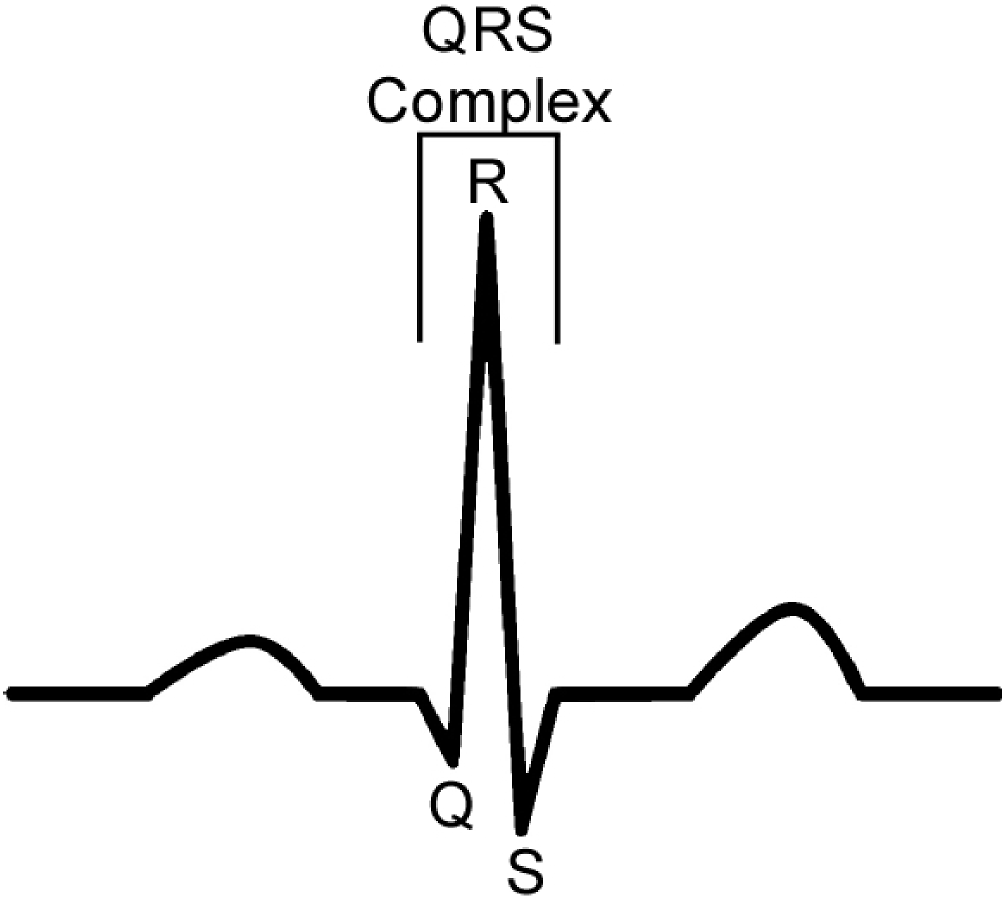


Figure 1.2: A simulated electrocardiogram waveform, illustrating the locations of the Q, R, and S inflection points which form the QRS complex.

per-application basis.

1.4 Example Application: Electrocardiograms

Within the domain of wearable devices, systems which monitor biological signals are very popular. Given their prevalence, it is natural to use a bio-signal monitoring device to bring in to context all of the background material presented in this chapter. Our example device will focus on electrocardiogram signals. But we will consider an application which traditionally requires some post-processing in the digital domain – the capture of the QRS complex.

While an analysis of the complete electrocardiogram (ECG) waveform provides the most data, a great deal of clinically relevant data can be gleaned from just the QRS complex. QRS complexes are useful for a variety of medical purposes ranging from monitoring for hyperkalemia or cardiac hypertrophy to simply extracting heart-beat to estimate perceived exertion [27].

In a traditional system, the analysis of a QRS waveform would begin by utilizing an ADC at a sampling rate dictated by Shannon-Nyquist sampling. Given that the highest rate of change in the QRS waveform of a 60-100 beats per minute heart rate occurs at a frequency of 57 Hz, the accurate reconstruction of the waveform requires a 114 Hz sampling rate. This 114 Hz sampling will result in 68-114 samples per QRS complex.

These 68-114 samples must then be processed by a microcontroller and/or transmitted by a transceiver. The total number of samples could be reduced by using an analog front-end to trigger sampling only when the heart rate is outside of a normal window [26], but this would still result in 68-114 samples per ‘interesting’ QRS complex. If you consider that this complex is only defined by three requisite data points, then this traditional method results in a thirty eight times potential oversampling. A better method would be to sample these three requisite points directly – avoiding the need of a microcontroller for digital processing and mitigating the use of a transceiver, resulting in system-wide power savings. Achieving this type of data-driven adaptive sampling is the goal of this work.

1.5 Summary

Now that the context of traditional sampling within sensor-interfaced devices has been set, this work can demonstrate a potential improvement – adaptive sampling for energy-constrained systems. Chapter 2 will begin by reviewing the requirements of, and the current literature on, adaptive and asynchronous sampling. Chapter 3 will present a SPICE compatible non-volatile analog-memory macromodel while Chapter 4 will present a frequency adaptive sampling scheme. The next half of this work will leverage these mathematical and simulation-based tools to develop full systems realized in physical circuitry. Chapter 5 will utilize the analog-memory model to create a temperature compensation circuit – this will provide a stable platform for data conversion to occur. Chapter 6 will discuss the use of adaptive sampling within the context of specific, bio-wearable related applications. I will extend this adaptive sampling method in Chapter 7 to outline an ADC architecture which takes further advantage of analog memory and the frequency-adaptive sampling method. Next, in Chapter 8, I will demonstrate the versatility of some of the developed techniques on an entirely different data-converter topology. Finally, Chapter 9 will summarize the key findings of this work and detail possible directions in which this research could be further developed.

Chapter 2

Technical Challenges in Asynchronous Quantization

Many energy-constrained sensor-interfaced devices (e.g., WSN nodes, bio-wearables) focus on the capture and analysis of 'bursty' signals. A bursty signal is one which is characterized by long portions of relative inactivity interspersed with shorter periods of high-frequency events. Systems tasked with monitoring bursty signals such as electrocardiogram (ECG), electromyography (EMG), or vocal signals are traditionally required to sample at a constant rate of twice the signal's highest frequency component – known as the Nyquist rate. This fixed sampling rate leads to oversampling during lower-frequency portions of the signal. This oversampling is even more wasteful in applications which do not require a perfect reconstruction of the signal, but instead are attempting to glean some specific information or data point.

Adapting to a signal's changing frequency characteristics or capturing specific points of interest is the primary goal of adaptive and asynchronous sampling systems. This chapter begins with a brief discussion of asynchronous and adaptive conversion systems in Section 2.1. One key difference is the mechanism which triggers these asynchronous systems, which is usually some sort of analog front-end. Section 2.2 will introduce the concept of using reconfigurable analog to alleviate some of the issues associated with using analog-front ends. Finally, Section 2.3 will provide some background on the issue of inter-sample time measurement – a unique challenge in asynchronous quantization.

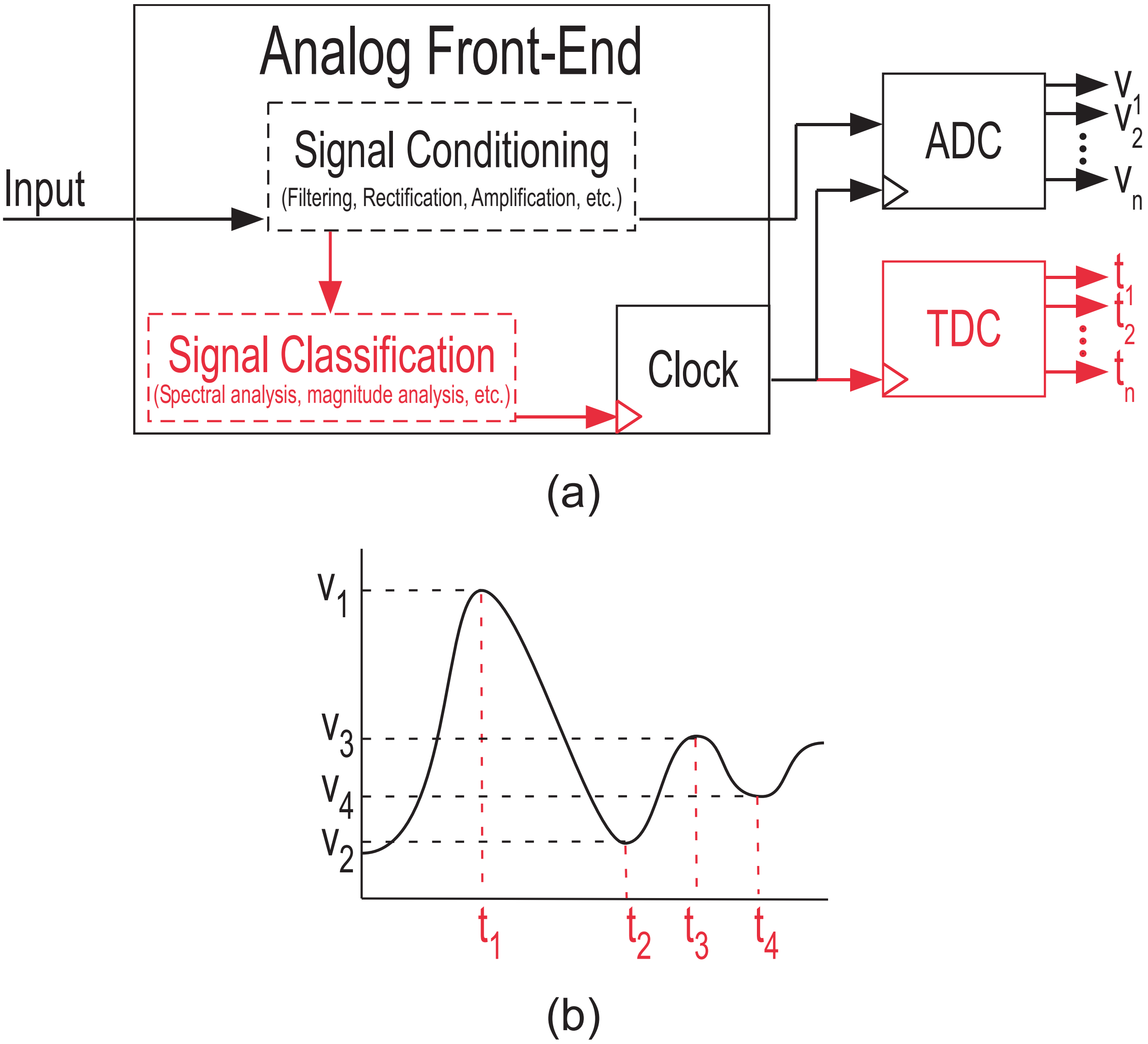


Figure 2.1: (a) A block diagram of a quantization system. The pieces shown in black are common to all quantization systems. The pieces shown in red are unique requirements of asynchronous quantization. The most important distinctions are the explicit time measurement, as well as the signal characteristic-driven clock. (b) A waveform showing example quantization levels. In an asynchronous system, the timing of the samples must be explicitly measured.

2.1 Asynchronous/Adaptive Analog-to-Digital Conversion

Besides improving the per-conversion-cycle efficiency of an ADC, the other logical approach to lowering the energy consumption of an ADC is reducing the number of conversions it performs. Asynchronous ADCs are a class of ADCs which are not clocked at the Nyquist rate, but are instead triggered by some external event – typically some characteristic of the target signal to be converted (Fig. 2.1). These ADCs push against the rate minimum imposed by the Nyquist Rate.

Compressed sampling, for example, is a sparse-signal measurement modality [28, 29] that has been implemented successfully as an asynchronous sampling scheme [30]. Compressed sensing works by creating an array of samples (taken from individual sensors) from a sparse signal. Given that the signal is very sparse, the samples can be represented by a vector of coefficients which are mostly zeros. The sampling rate in this scheme is much smaller than the dimension of the signal being measured, meaning that the quantized array of samples would normally not be sufficient to reconstruct the original signal. However, the additional information that the signal is sparse allows the signal to be reconstructed [28].

Level-crossing ADCs are possibly the most popular type of asynchronous ADCs. The basic idea is that a sample is only converted when the measured signal passes through a bound that would represent a new digital word. By only recording these transitions, periods of relative inactivity are ignored. Ignoring these inactive periods allows energy to be saved through reducing the number of conversions [31]. While this method avoids extraneous conversions during periods of low-activity, it is possible for it to cause oversampling during 'bursty' events. If an event is characterized by a pulse of constant derivative, an electrocardiogram spike for example, it may trigger several threshold crossings. However, at least in the case of constant derivative change, accurate reconstruction could be achieved through the use of only the first and last threshold crossing, making the intermediary crossings extraneous information. Another way to consider this issue, is that as the resolution of the ADC is increased, the frequency bandwidth becomes more limited.

Extrema sampling, otherwise known as peak or min/max sampling, is another implementation of an asynchronous sampling method. Here, only the local minimums and maximums of a signal are sampled. This method is similar to level-crossing sampling in that it adapts itself to the changing spectrum of a signal, but this method leads to fewer sample points. Fewer sample points can mean a loss in signal reconstruction fidelity, but it can also mean a reduction in superfluous samples that may occur in level-crossing sampling when a function is rapidly changing with a near constant rate of change [32, 24].

2.2 Reconfigurable Analog Front Ends

As described in Chapter 1, signal processing in the analog stages is very energy efficient, but carries with it some inherent limitations. Chief among these limitations is the application-specific nature of analog electronics. While an analog front-end might make an excellent conditioning or classification circuit to trigger an asynchronous data converter, it would likely be useful for only a narrow range of applications. For this front-end to be useful in a wide range of applications, particularly for triggering asynchronous data conversion, a certain amount of tunability or reconfigurability is necessary.

The desire to enable reconfigurability in analog electronics has led to the development of a relatively new class of devices, field-programmable analog arrays (FPAAs) [33, 34, 35, 36, 26, 26, 37, 38, 39]. FPAAs are similar to digital field-programmable gate arrays (FPGAs) in that they allow for a system architect to program arbitrary connections of primitives to form larger systems.

FPAA usage varies from trying to solve a single-niche problem, to implementations which try to provide reconfigurability in analog at a level comparable to what FPGAs provide in digital. Ultra focused solutions include an FPAA which is focused on the implementation of a pipeline ADC [40]. In this case, the FPAA allows for both tunable pre-quantization filtering as well as some reconfigurability in the actual quantization stage itself, theoretically enabling the implementation of an entirely different topology – a sigma-delta converter. On the other end of the spectrum is a system that allows reconfigurability at the transistor level [36]. This level of reconfiguarbility allows a great number of analog designs to be useful, thus making it a useful tool for prototyping or teaching, among other applications.

Somewhere along this spectrum are FPAAs designed for use within energy-constrained applications [26]. These FPAAs implement analog blocks of varying granularity – from single transistor amplifiers up to oscillators and operational amplifiers. This range of granularity strikes a balance between the performance of individual blocks and the reconfigurability of the system as a whole. This variety in available circuitry also allows these systems to focus on signal classification and event-detection as opposed to simply signal conditioning (e.g., filtering operations). A great deal of the flexibility in both topology and biasing for this and other FPAAs is provided by non-volatile analog memory, otherwise known as floating-gate transistors.

2.2.1 Floating-Gate Transistors

Floating-gate transistors are field-effect transistors characterized by their floating control-gate node. This node has no DC path to ground, and instead relies on capacitive coupling and charge programming to effect the gate potential. These devices are most widely used in digital flash-memory applications, but they have also been shown to be extremely useful in the analog domain. One use of floating-gates is as non-volatile analog memory, in which they are programmed to store a particular charge. By tuning the charge stored on the floating node, they have been used for offset removal in differential pairs [41], threshold definition in flash ADCs [42], or for tuning corner frequencies in filters [43]. The versatility of floating-gate circuits has allowed them to be useful in applications ranging from modeling neurons [44] to measuring cranial impacts caused during collisions in sports [3].

While floating-gate devices are very useful, they are also uniquely challenging to implement. This difficulty largely stems from the fact that the two main sources of programming, Fowler-Nordheim tunneling [45] and hot-electron injection [46], occur with currents on the order of pico-Amperes. This complication is only worsened by the SPICE circuit simulation language's inability to accurately model these device due to their floating-nodes.

2.3 Inter-Sample Time Quantization

Unlike traditional ADCs, the asynchronous variety have an unknown and variable amount of time between samples. What is often ignored within the literature, however, is that proper knowledge of this time period is critical for proper reconstruction of the signal [47].

Most of the current literature ignores this timing issue completely [48, 49]. Others utilize a stand-alone data acquisition system to separately monitor the time limits for the sake of prototyping the system [24]. [50] proposes that, instead of explicitly measuring the time

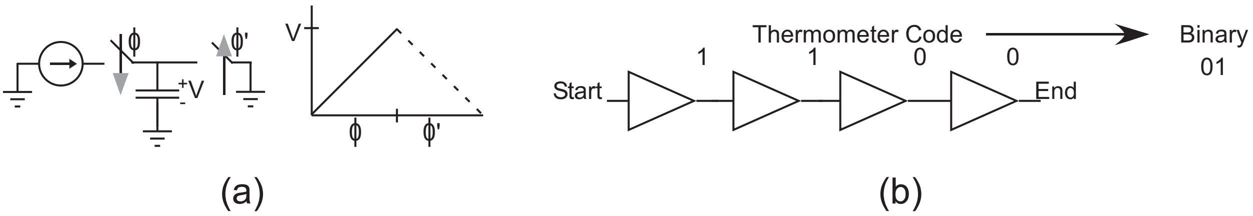


Figure 2.2: (a) A simplified schematic and output of a class-1 TDC. This type of TDC uses a voltage ramp to create a quantity that is proportional to the duration for which it is charged. (b) A simplified schematic and output of a class-2 TDC. This type of TDC uses equal delay units to create a thermometer code which can be converted to binary.

interval of an asynchronous ADC deployed at a WSN node, the time could be interpolated by broadcasting every sample as it is acquired and the base station could measure the time in between samples. This method introduces an entire host of variables involved with the transmission and reception of these signals – making this method very impractical.

A more practical approach would be to use an explicit time to digital converter (TDC). TDCs measure temporal characteristics within a system and convert that quantity to a digital value. TDCs are used in various digital oscilloscopes, they are inherently found in delay-locked loops, and they also appear in positron emission tomography scans (PET scans). Within the context of PET scans, TDCs are used to measure the duration of the annihilation of a positron emitting radioisotope (injected in a subjects body) with an electron. [51] was a sub-nano second TDC developed for the express purpose of use in PET scans.

The class-1 TDCs operate much like an integrating ADC (Fig. 2.2 (a)). In this class, a current source creates a ramping voltage upon a capacitor. This ramp begins at a known voltage and is halted at the end of the period to be measured. The resulting analog voltage is thus proportional to the duration of the asynchronous event and can be converted to a digital word through any traditional ADC technique.

Class-1 TDCs have been utilized in works which require exceptionally low power operation and demand small area [24], but have several severe drawbacks. Like integrating ADCs, class-1 TDCs are not particularly suited for CMOS implementation. It is inherently difficult to build a stable current source that will be reliable in the face of power supply variation or fabrication process variations. This technique also scales poorly as CMOS fabrication processes continue to shrink. Also, an entire ADC unit is required after this initial TDC stage to convert the analog value to a usable digital word.

The class-2 TDCs (Fig. 2.2 (b)) are most similar to digital delay lines. A start signal is passed to a chain of inverters, buffers, or other generic delay unit. The signal propagates along the line, clocking a flip flop or similar memory element at every node, until a stop signal is received. The result is a thermometer code which is proportional to the duration of the event and can be converted easily to a digital word.

Within class-2 TDCs, certain comparisons can be made to typical ADC design. For instance, the LSB of this circuit is simply the smallest time increment which can be measured – at best this is the smallest delay created by a single delay unit. Another example is instead of defining the dynamic range as the range of analog voltages which can be converted, we can define it as the maximum amount of time which can be measured. In theory, we can then extend the dynamic range to an arbitrary length by extending the delay-line to an arbitrary length. In reality, this technique is limited by non-linearities that either exist in asymmetries within the schematic design or the physical layout. Put another way, eventually the circuit will drift in timing past one least significant bit and some digital correction will be necessary.

Within conventional circuits, TDCs have found a few interesting uses. Time interleaved ADC structures are often used to increase the through-put of slower ADC topologies, but at some point the designer naturally hits a point where process mismatch becomes non-negligible and time-interleaving is no longer feasible. In [52], a TDC was used to automatically detect variation caused by process mismatch and push the resolution of the time-interleaved system past what it could perform at otherwise.

2.4 Summary

This chapter has introduced concepts critical to understanding the remainder of this work. These concepts begin by understanding the difference between asynchronous/adaptive conversion systems and synchronous conversion systems. The idea of alleviating the application specific nature of analog prepossessing with reconfigurable analog was then introduced as a mechanism for triggering the event-driven asynchronous conversion. Finally, the concept of inter-sample time quantization was discussed – an issue unique to asynchronous conversion.

Chapter 3

A SPICE-Compatible Floating-Gate Macromodel

Despite their usefulness as programmable circuit elements, floating-gate circuit development is hampered by its inability to be modeled in SPICE-compatible environments. The primary challenge in modeling a floating gate FET device is that there is no DC path to the floating node itself – a property which SPICE has difficulty reconciling at compile time. This lack of a DC path makes modeling capacitive coupling onto the node and effecting the charge stored on the floating node impossible without the use of a macromodel. Previous attempts at creating a macromodel have not addressed these issues simultaneously in a manner compatible with AC, DC, and transient analysis. In this chapter, we present a new SPICE-compatible macromodel which includes mechanisms for capacitive coupling onto the floating node as well as mechanisms for programming the charge on the floating node. In addition, this work provides insights into the design and use of floating gates and varies the validity of this model with floating-gate circuits developed over multiple processes.

3.1 Floating-Gate Modeling in SPICE

Floating-gate devices are most widely used in digital flash-memory applications, but they have also been shown to be extremely useful in the analog domain. One use of floating-gates is as non-volatile analog memory, in which they are programmed to store a particular charge.

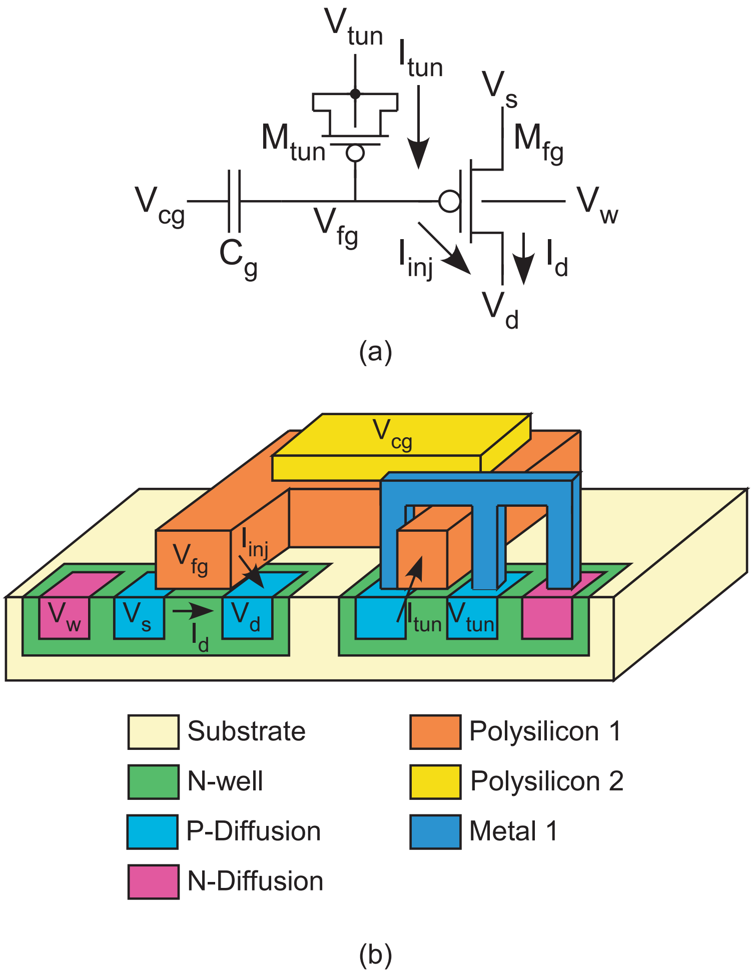


Figure 3.1: The basic structure of a pFET CMOS floating-gate element. (a) Schematic representation of a pFET floating-gate element. Note that the floating node, V_{fg}, is where charge is stored. It is the lack of a DC path to ground from this node which makes direct SPICE modeling impossible. (b) A model of the layout of a pFET floating-gate element. This layout topology has the advantage of being directly implementable in any double-polysilicon process, without the need for amended fabrication rules.

By tuning the charge stored on the floating node, they have been used in many applications including offset removal in differential pairs [41], threshold definition in flash ADCs [42], or for tuning corner frequencies in filters [43]. The versatility of floating-gate circuits has allowed them to be useful in applications ranging from modeling neurons [44] to measuring cranial impacts caused during collisions in sports [3]. Floating gates have even enabled the creation of field-programmable-analog-arrays, the analog equivalent of digital FPGAs, by providing programmable switches and bias currents [26, 36].

Despite the usefulness of floating-gate devices within a wide variety of applications, SPICE modeling techniques are incomplete and vary from application to application. The problem with SPICE modeling arises from the unique floating node, V_{fg} in Fig. 3.1(a), which has no DC path to ground. Without a DC path to ground, SPICE simulations are unable to both set and effect the charge on the node and maintain accurate effects of coupling from explicit and parasitic capacitances. Macromodels have been developed to address these issues, but they address either the charge setting and programming issue [53, 54, 55] or the capacitive coupling issue [56, 57], but never both. While these techniques can be accurate and adequate if applied to specific applications, a macro-model which addresses the complete behavior of a floating-gate would better equip circuit designers to predict the intended and unintended behavior of their systems. In addition, a unified floating-gate model promotes the use and re-use of floating-gate based circuits within other systems.

In this chapter, we will detail a unified macro-model for floating-gate devices which is capable of modeling the DC capacitive coupling characteristics as well as the storage and programming of charge on the floating-node. This work expands and improves upon our previous work [58] by improving the macromodel (more accurate charge modeling equations and adjustments which improves convergence) and by providing better instruction on its implementation as well as better verification for its use across multiple fabrication processes. We will begin by explaining the structure of the floating-gate element in Section 3.2. Next, the theory behind the mechanisms of charge modification and capacitive coupling will be explored in Section 3.3. We will then present the macromodel itself in Section 3.4 before describing empirical and analytical methods of parameter extraction in Section 3.5. Finally, we will provide verification of the static and dynamic characteristics of this model by comparing it to designs fabricated over different fabrication processes in Section 3.6 before concluding in Section 3.7. All simulated data will be compared to real data from devices fabricated in 0.5μm and 0.35μm standard CMOS processes.

3.2 The Floating Gate Structure

Floating-gate elements are most simply characterized as a MOSFET whose gate lacks an explicit resistive path to ground. Instead, a ‘control gate’ is capacitively coupled onto the

DC isolated node, or 'floating gate' (FG). This FG node stores charge, making it not only useful for digital flash memory, but also for non-volatile reprogrammable analog memory.

Figure 3.1(b) shows the physical layout of a floating gate device. Alternative layouts such as nFET type FG elements, which use double well processes, or stacked capacitor FG elements are realizable, but the demonstrated design can be made in the most common CMOS fabrication processes.

The FG itself is formed through the use of a polysilicon-insulator-polysilicon capacitor. If a double polysilicon process is not available, this structure could still be realized with a MOS capacitor – trading some ease of analog design for some space reduction. When drawing the layout of this particular node, it is best-practice to reduce charge leakage by not creating any connections to metal paths. Instead, it is best to make the connection to the tunneling capacitor via polysilicon, as depicted. The connection of the floating node to nonessential terminals or gratuitous sizing of traces, even within the polysilicon layer, has been reported to cause poor charge retention, or 'leaky' FG nodes – as in the case of [59] where this effect was intentional.

Similar to a standard MOSFET, the channel current of an FG transistor is determined primarily by the potential between the source and gate over top of its channel. What makes the FG transistor unlike a standard MOSFET is the fact that the potential on the gate is determined by both the charge stored on the FG node, as well as any voltages capacitively coupled onto the node. If the charge on the FG is Q and the total capacitance on the FG is CT, then

$$V_{fg} = \frac{Q + C_{cg}V_{cg} + \sum C_x V_x}{C_T} \tag{3.1}$$

where C_{cg} and V_{cg} are the drawn capacitance from the FG node to the control node and the voltage placed upon the control node respectively, and C_x and V_x are any other drawn or parasitic capcitances to the floating node (e.g. drain-to gate capacitance, source-to-gate capacitance, etc.) and the voltage placed upon them. Typically, the drawn control gate capacitance is much larger than the other capacitances, allowing reasonable approximation with simply the C_{cg}, V_{cg}, and C_T terms, if computational speed is preferred over accuracy.

Even for the case where we ignore parasitic capacitances and consider only drawn control-gate capacitances (of which there can be more than one), we can begin to see some interesting properties of FG transistors which can be taken advantage of in the analog domain. The first and most obvious is that the charge placed upon the FG creates a programmable, non-volatile offset. Another way to think about this programmed charge, is to consider it as repositioning the effective threshold voltage (at least from the perspective of a circuit using the floating gate as a black-box). Another advantage is that, particularly with multiple control gates, the FG transistor can be used as a multiple input transistor [60], possibly also taking advantage of the capacitive division to create inputs scaled to particular ratios [61].

3.3 Reactions of the Floating Node

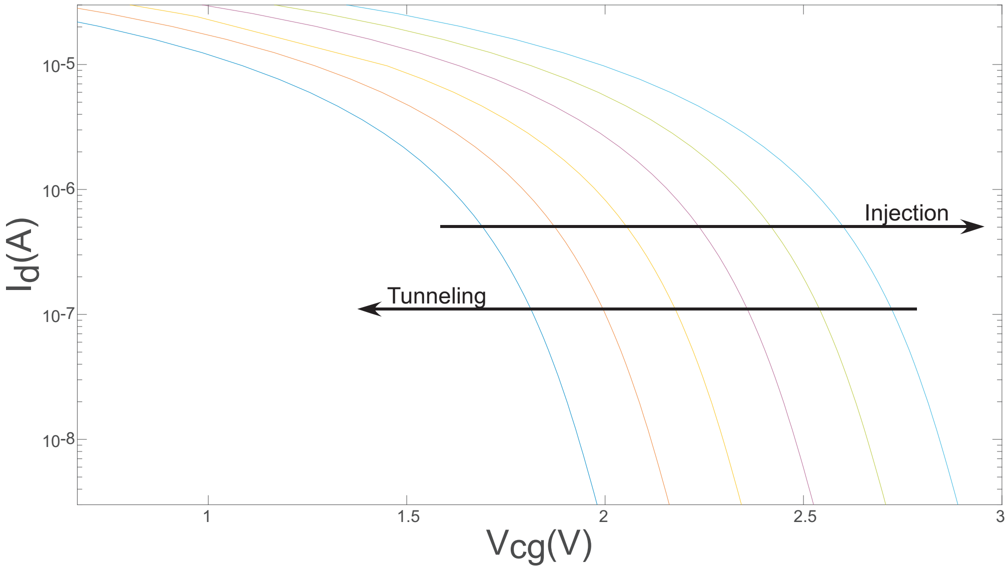


Figure 3.2: The use of Folwer-Norheim tunneling and hot-electron injection allow circuit designers to program FG elements with specific charge – in effect, the threshold voltage can be 'tuned.'

The ability to effect and predict the charge on the floating-node is of paramount importance for the usage of FG devices. Charge is typically programmed through Fowler-Nordheim tunneling and hot-electron injection (Fig. 3.2). For a model to be accurate, it must also account for methods of changing the charge which are typically unintentional or undesired, specifically leakage and a previously un-discussed phenomena we call reverse-tunneling. In this section, we will describe the effects of these mechanisms and the mathematical equations which model them.

3.3.1 Tunneling

Fowler-Nordheim (FN) tunneling is the primary method by which charge is removed from the floating node. FN tunneling can be used for writing or erasure in individual FG elements. For larger arrays of FG elements, it isn't normally applied to individual elements, instead it is most commonly used for global erasure. The reason is due to the relatively high voltages required to induce the phenomena. This phenomena occurs when high voltages are used to distort the energy band of the oxide, allowing electrons to pass through. Generating this high voltage for most systems is a source of inefficiency, particularly if it must be done on a per/write basis.

FN tunneling can be performed on FG nodes made of pFET or nFET transistors. However, we recommend the use of pFet devices for a few reasons. The first reason is that, to protect against reverse breakdown, tunneling junctions are normally isolated by placing them inside a well. Therefore, the use of an nFET necessitates a double-well process. Other reasons are that we have found the pFET tunneling junctions to produce higher levels of current, enabling faster erasure. Also, their erasure time is not voltage dependent. We recommend sizing this tunneling junctions at about 3.1 percent of the total capacitance (drawn and parasitic) of the FG element to maximize current draw while minimizing voltage coupling and oxide degradation. Further elucidation and analysis of these points can be found in [62].

The current induced by FN tunneling can accurately be modeled by the equation:

$$I_{tun} = \alpha(\frac{C_{tun}}{\gamma})exp[-\frac{\beta t_{ox}}{1-(Ctun/C_T)V_{tun}-V_{fg,e}}] \tag{3.2}$$

where C_{tun} is the capacitance of the tunneling junction, C_T is the total explicit and implicit capacitance seen by the floating node, t_{ox} is the thickness of the oxide barrier, V_{ox} is the voltage across the barrier, γ is the unit capacitance (aF/m^2) of C_{tun}, and α β are constants related to the fabrication process and junction type [45, 63]. By curve-fitting FN tunneling data from multiple chips and processes, we have found that values of 185.5 A/m^2 and 32.8

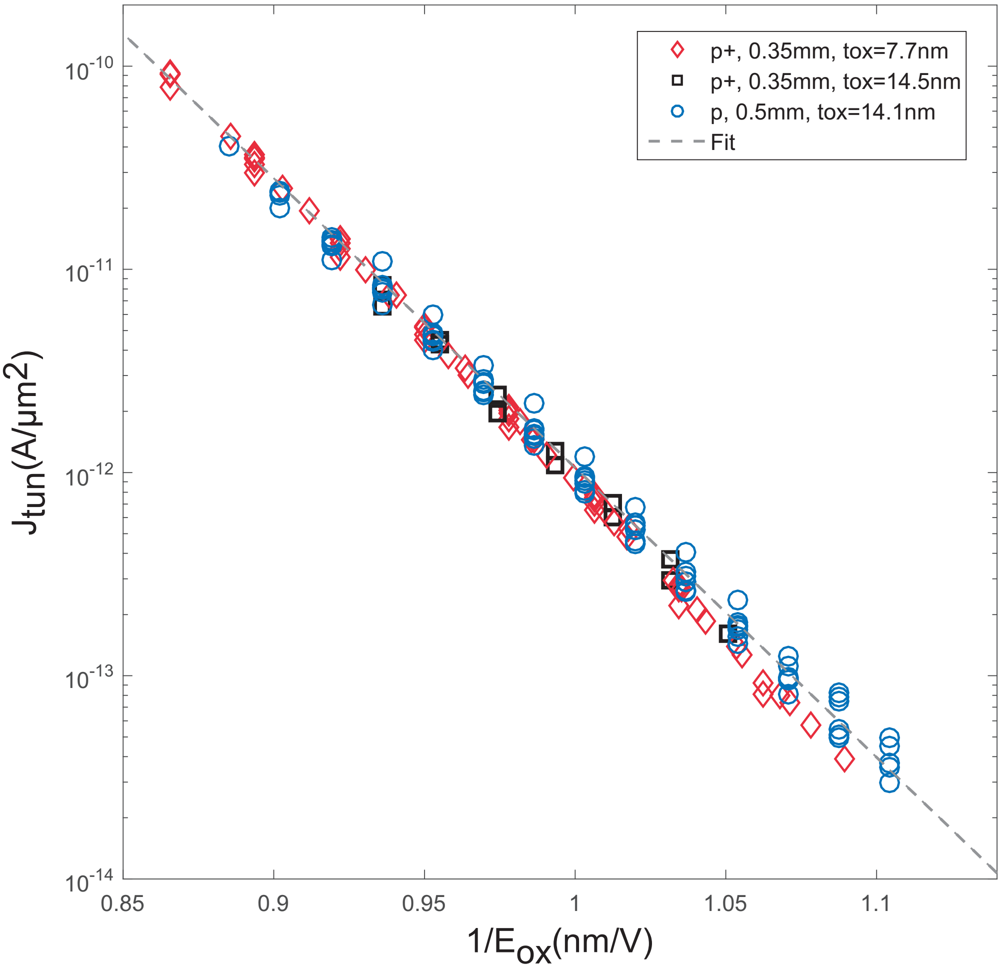


Figure 3.3: Data taken from multiple processes were used to derive the constants used in the calculation of tunneling current.

V/nm, for α and β respectively, accurately model the current induced by injection across a range of fabricated chips [62] (Fig. 3.3).

3.3.2 Injection

Hot-electron injection is the method by which charge is added to the FG element. Hot-electron injection occurs when there is a channel current flowing through the transistor and a drain-to-source potential high enough to cause impact ionization at the drain [46]. A portion of these ionized carriers gain enough energy to pass through the oxide barrier and become trapped on the floating gate. The resultant injection current can be approximated, in the

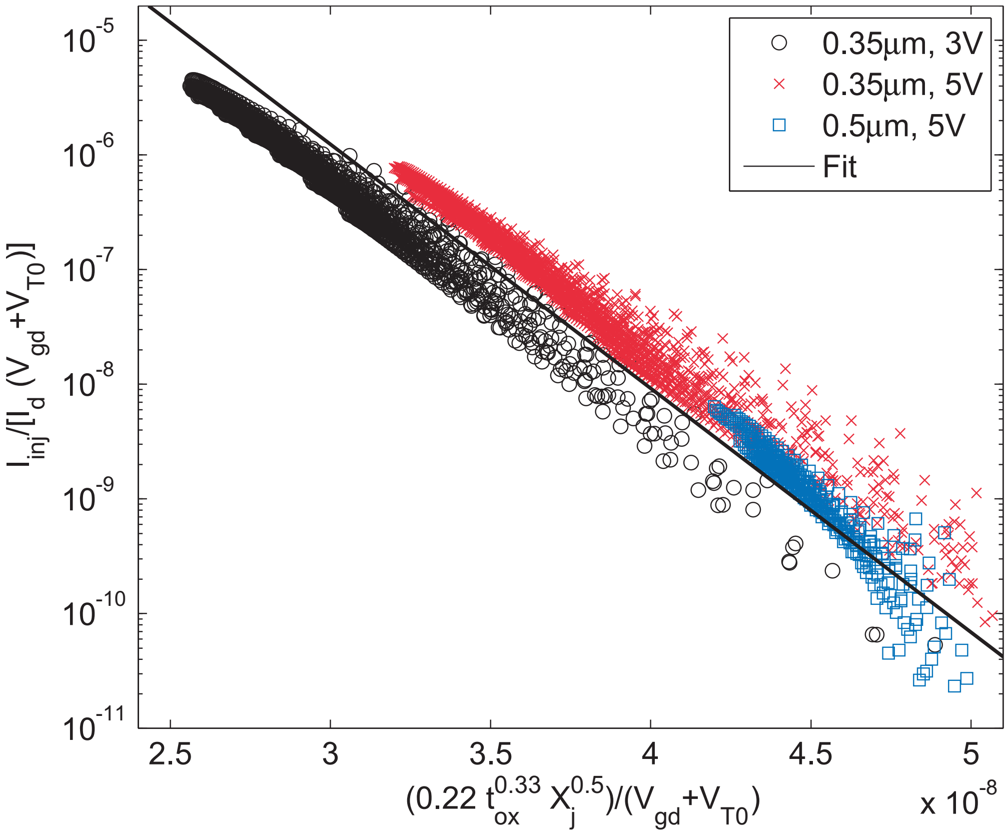


Figure 3.4: Calculated injection rates from this experiment were normalized across processes to find the constants we use in our injection current calculation.

subthreshold region, as:

$$I_{inj} = \gamma * I_d * (V_{fg} - V_d + V_t) * exp[-\frac{\gamma\delta}{(V_{fg} - V_d + V_t)}] \tag{3.3}$$

where I_d is the channel current, V_{fg}, V_d, and V_t are the floating gate, drain, and threshold voltages, and γ δ are device-dependent fits found to be of value 3 and 4.9e8, respectively. [64]. This current can be modeled as flowing from the floating-gate to the drain, resulting in a lowering of the floating-gate potential.

Injection is typically a more practical means of programming compared to tunneling, do to the lower required potential ($V_{sd} > 3.5$V vs $V_{tun} > 8$V for 0.35μm $CMOS$ process). There are generally two categories of injection programming methods: pulsed and continuous time. In pulsed based programming, short pulses of injection are induced with the value of the floating node measured after each pulse [65, 66]. While this method is very accurate, it is also very slow due to the need for read cycles after every pulse. Continuous-time programming, conversely, implements a feedback structure to stop programming of the FG eleement once it reaches the desired value. Generally, continuous-time programming is quicker and requires less peripheral circuitry [67].

3.3.3 Reverse-Tunneling

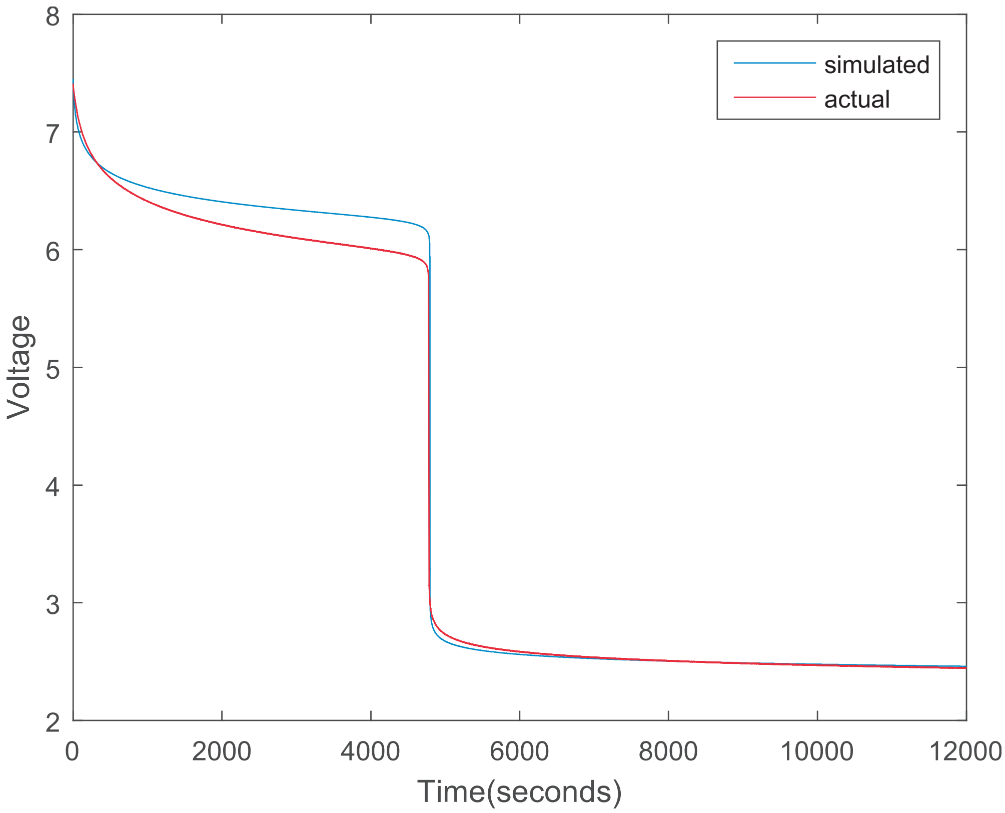


Figure 3.5: Demonstration of the effect of reverse tunneling. In this test, the source of the FG element was held at 6.5 volts, and the FG node was programmed with sufficient charge so that it would be well above the drain. With the FG node held high, we would expect there to be no channel current and consequently no injection current. However, we can see an injection-like effect occurring over a very long time scale. We attribute this to reverse tunneling.

Reverse tunneling is a yet unstudied phenomena that resembles injection at first approximation. This phenomena occurs when the potential of the floating node is sufficiently higher than that placed on the drain node, causing charge to tunnel from the floating node to the drain node. As can be seen in Fig. 3.5, the process is fairly slow, due to the very slight magnitude of current involved. We have found that scaling the Fowler-Nordheim tunneling equation, Eq. 3.2, to one half of a percent is sufficient for accurate modeling.

In practice, this may occur during programming of an array of FG elements. After writing a value to a FG node using injection, it is common practice to raise the potential of the control gate before moving onto the next FG element, with the intention of completely shutting off the channel current to avoid further programming. However, doing so has the potential to raise the FG node to a high enough voltage to induce reverse tunneling. If this effect is not properly understood and modeled, it may be mistaken for some non-ideality such as charge leakage or some unintended reprogramming through injection.

3.3.4 Charge Leakage

The term 'charge leakage' is used to describe the effect that a, usually unintentional, DC path to the floating node can have. This DC path is usually formed one of two ways. The first and most common is through ill-advised layout practices, contrary to those suggested in Sec 3.2.

The second possible cause of a DC path is oxide degradation. The primary methods of charge manipulation on an FG device, namely injection and tunneling, are inherently destructive processes which have a slight, degrading effect upon the oxide. However, it has been shown in both the digital and analog domains that excellent charge retention can be maintained with loss less than one percent over a decade of use [68]. Thus, for the majority of applications, circuit designers can rely on consistent performance from FG devices, provided that some basic best practices were followed in the layout stage.

3.4 Floating-Gate Macromodel

In modeling the floating gate (FG) transistor, as seen in Fig. 3.6, the three details addressed by our model are: modeling the charge stored on the floating node, modeling the means by which charge may be programmed, and modeling the effects of capacitive coupling on the floating node. While previous macromodels have been able to model these effects individually [53, 54, 55, 56, 57], ours is capable of modeling all of these effects in tandem.

The simplest part of the model to understand is the capacitive coupling. Changing the

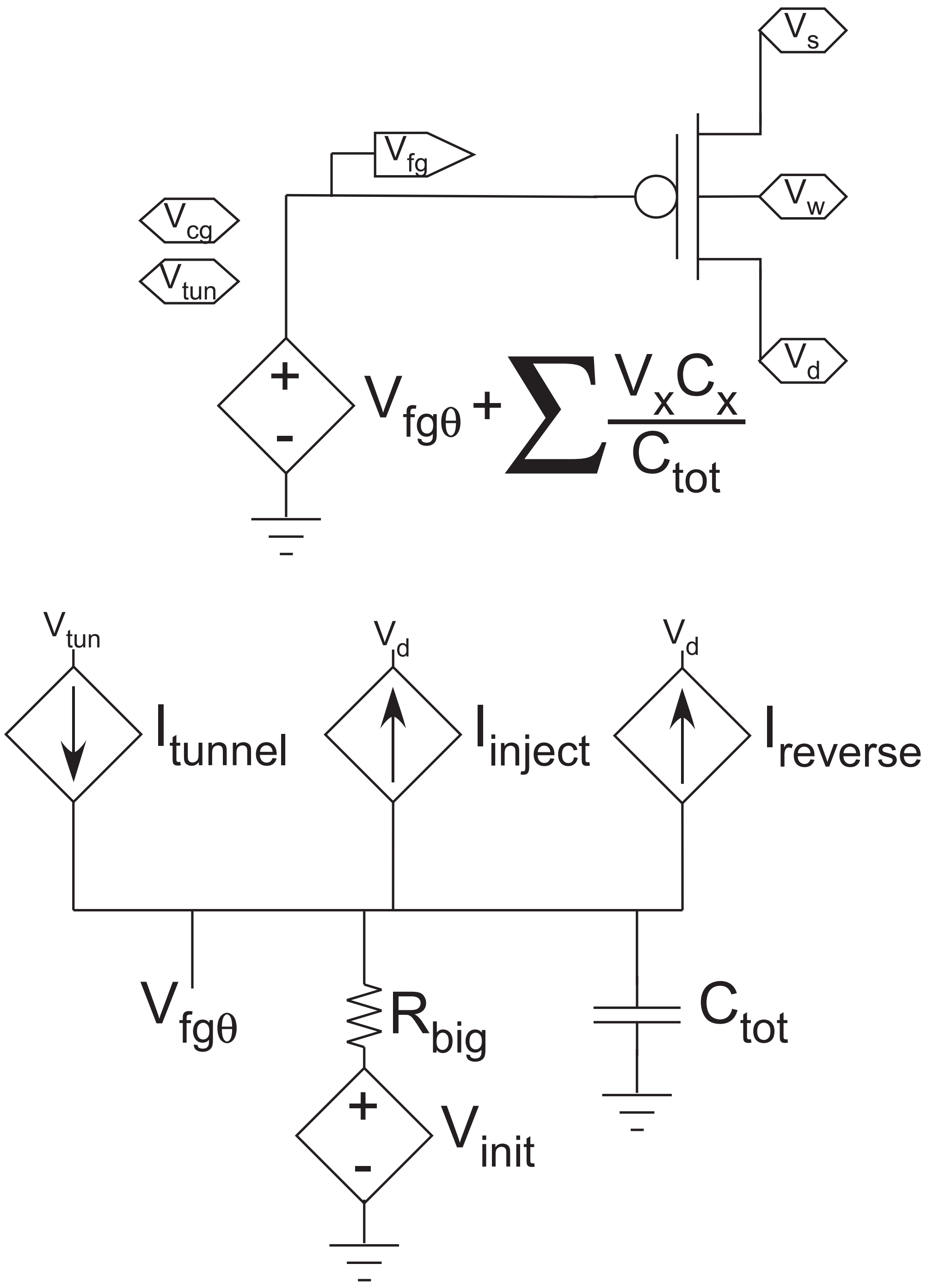


Figure 3.6: Schematic representation of our floating-gate SPICE model. The bottom portion reflects charge programming techniques as well as non-idealities which may effect charge. This node is combined with the effects of capacitive coupling onto the gate of the transistor in the upper portion of the schematic.

voltage on the floating node through capacitive coupling is achieved through a dependent source attached directly to the floating node.

The dependent source should sum all capacitively coupled potentials individually, through the rule of capacitive division. A complete model of this should include the contributions of all drawn and parasitic capacitance (e.g., drain-to-gate capacitance, source-to-gate capacitance, etc.) but, in practice, the drawn coupling capacitance tends to dominate. Therefore, if computational speed were desired, as in the case of simulating a large array of FG-enabled circuits, one could include only the drawn capacitance and achieve reasonable accuracy.

The other term added through the dependent source is the voltage seen at the ‘dummy’ node $V_{fg\theta}$. This node is used to both set the initial voltage as well as to model the effects of charge programming. The initial voltage is set by an independent source attached to $V_{fg\theta}$ through a very large resistance. This large resistance allows $V_{fg\theta}$ to reach the desired value prior to a transient simulation, when the SPICE simulator is searching for a stable DC operating point, but then effectively separates the voltage source's effect during simulation. One caveat to this operation is that one must remember that the capacitively coupled voltages are also being added in during the search for a starting DC operating point. Therefore, the equation to set this initial value should be

$$V_{init} = V_{des} - \frac{\sum C_x V_x}{C_T} \tag{3.4}$$

where V_{des} is the desired initial floating gate voltages. Alternatively, if one would rather think of the programming of a floating gate as a shifting of threshold voltages, then the voltage source should be set to

$$V_{init} = V_{Tact} - V_{Tdes}\frac{C_{cg}V_{cg}}{C_T} - \frac{\sum C_x V_x}{C_T} \tag{3.5}$$

where V_{Tdes} is the desired threshold voltage and V_{Tact} is the actual, intrinsic, threshold voltage.

Also seen at the dummy node are current sources and a capacitor. The capacitor, set at the value of the total capacitance seen by the FG node, and the current sources create the voltage changes that charge reprogramming would have on the FG node. Included in the illustration are sources which model injection, tunneling, and reverse tunneling (see Sec. 3.3 for specific equations). In addition to these effects, other charge reprogramming effects could be added to model new behaviors which may arise from newer processes or particularly odd topologies.

3.4.1 Verilog-A Implementation

In certain cases, it may be beneficial to use a SPICE language which does not support the use of arbitrary voltage or current sources. In these cases, it is necessary to model the injection, tunneling, and other charge programming currents with a high-level programming language. The following is the implementation of the tunneling current in Verilog-A:

```
‘include "constants.vams"
‘include "disciplines.vams"

module TunnelingMod(Vtun,Vfg,Vfg0,Test);
inout Vtun,Vfg,Vfg0,Test;

electrical Vtun,Vfg,Vfg0,Test;

real tun_val;

parameter real tdelay=0 from [0:inf);
parameter real trise=2n from [0:inf);
parameter real tfall=2n from [0:inf);

parameter aT=185.5e12;
parameter bT=32.8e9;
parameter Width=2e-6;
parameter Length=1e-6;
parameter tox=7.754e-9;

analog begin
```

```
        @(initial_step)
        begin
                tun_val=0;
        end

        if(analysis("ic"))
                tun_val=0;
        else begin
                if(V(Vtun,Vfg)>8)
                    tun_val=aT*Width*Length*exp(-tox*bT...
                        /(V(Vtun,Vfg)));
                else
                    tun_val=0;
        end

        I(Vtun,Vfg0) <+ transition(tun_val, tdelay, trise,...
            tfall);

        V(Test) <+ transition(tun_val, tdelay, trise, tfall);
end



endmodule
```

Likewise it is possible to model injection in Verilog-A. Implementation of this equation in Verilog-A takes the following form:

```
‘include "constants.vams"
‘include "disciplines.vams"

module InjectionMod(Vd,Vs,Vfg0,Vfg,Vcur,Test);
inout Vd,Vs,Vfg,Vfg0,Vcur,Test;

```

```
electrical Vd,Vs,Vfg,Vfg0,Vcur,Test;

real inj_val;
real alpha;
real velLen;
real beta;

parameter real tdelay=0 from [0:inf);
parameter real trise=2n from [0:inf);
parameter real tfall=2n from [0:inf);

parameter gamma=3;
parameter delta=4.9e8;
parameter xj=3e-7;
parameter tox=7.754e-9;
parameter Vt=0.6915;

analog begin

        alpha=gamma;
        velLen=0.22*pow(tox,0.333)*pow(xj,0.5);
        beta=delta*velLen;



        @(initial_step)
        begin
                inj_val=0;
        end

        if(analysis("ic"))
                inj_val=0;
        else begin
                if(V(Vs,Vd)>3.5)
                    inj_val=(alpha*V(Vcur)*V(Vfg,Vd)+Vt)...
                        *exp(-beta/(V(Vfg,Vd)+Vt));
                else
                    inj_val=0;
        end

        I(Vfg0,Vd) <+ transition(inj_val, tdelay, trise,...
            tfall);

        V(Test) <+ transition(inj_val, tdelay, trise, tfall);
end



endmodule
```

In both cases, an important consideration is determining a pre-transient analysis, or DC, operating point for the charge programming currents. The most straight-forward approach to this consideration, is to initialize them to a value of zero. Almost all other approaches will cause the simulator to assume that, for example, the floating-gate has been injecting indefinitely prior to the transient analysis. This faulty assumption will usually result in either a simulation failure or an erroneous result.

In order to set the current sources which model charge reprogramming, certain values from the target CMOS library must be explicitly included. These include threshold voltage, oxide thickness, and junction depth. For complete agreement with fabricated results, these values may need to be 'tuned' to match the actual values of the fabricated device (as described in Sec. 3.5).

3.5 Tuning the Floating-Gate Model

The first step in implementing the floating-gate macromodel is tuning the model to fit the fabricated device. The process amounts to finding the threshold voltage, oxide thickness, and junction depth through the foundry reports available for individual fabrications. These parameters, along with the multi-process parameters presented in Sec. 3.3, can be used to model injection and tunneling currents.

The other necessary component to be analytically derived are the drawn and parasitic capacitances. The coupling capacitance will be by far the largest of these, and is therefore the most crucial to calculate accurately. Foundry reports on the overlap capacitance and area between polysilicon layers can be used to derive this value. Parasitic capacitances can be estimated by the width of the transistor and the reported value of overlap capacitance.

Beyond using foundry reported values, further refinements can be made to tune the model to a given process or even to a specific fabrication run. This section will detail how we characterized the charge programming effects as well as how we tuned the model for specific ICs.

3.5.1 Calculation of Injection and Tunneling Constants

Calculations of the tunneling and injection constants, provided in Sec. 3.3, were made on data taken from multiple processes. These calculations were made by data taken from the circuit in Fig. 3.7. This circuit provides a feedback loop, which was used to measure the effects of both tunneling and injection. The basic operation of the circuit consists of inducing the charge programming effect (i.e., tunneling or injection) and observing the feedback loop's effect on the control-gate while also monitoring the channel current. Injection can be induced with a sufficiently high drain-to-source voltage while tunneling can be induced with a sufficiently high tunneling voltage. The feed-back loop will then cause the control gate to change to compensate for the changing voltage on the floating-node. This effective slope in control gate voltage is then proportional to the injection/tunneling current by a factor of C_{cg}.

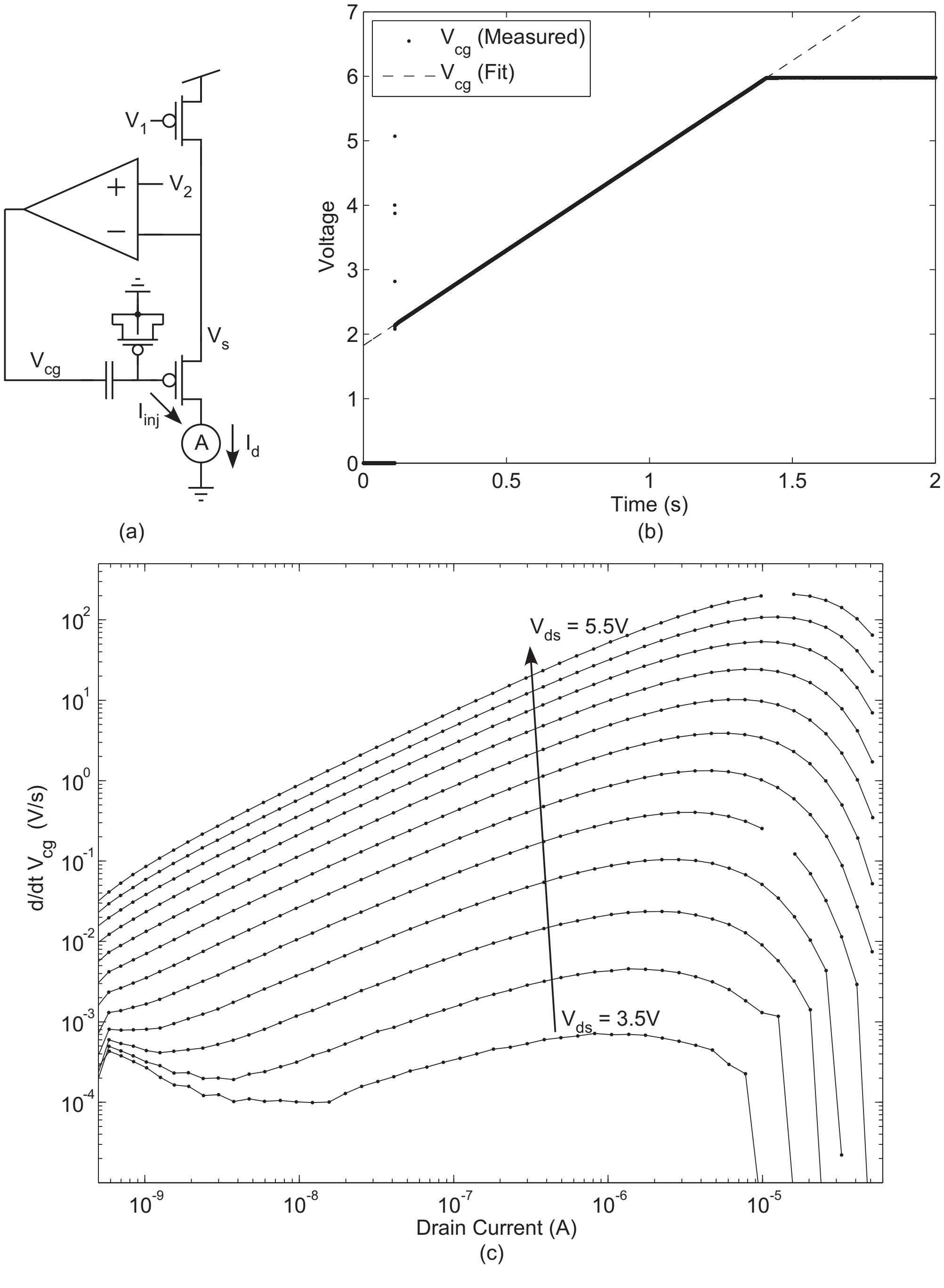


Figure 3.7: Demonstration of acquisition of injection data. (a) This continuous-time programming circuit was used to inject or tunnel the FG element at a set source potential and channel current value. (b) The results of one injection programming. Here we can see the measured V_{cg} raising at a constant rate to compensate for the lowering of the floating node potential through injection. Measurement of the slope, along with knowledge of the coupling capacitance from the control gate, allows us to calculate the value of injection for this single value of channel current and source voltage. (c) The aggregate of multiple injection programmings for many channel currents and source potentials.

3.5.2 Fine-Tuning the Model for Fabricated Devices

Using the device values found in fabrication-specific foundry reports and the charge programming constants presented in Sec. 3.3 will provide reasonably accurate results, but these results can be refined further for individually fabricated ICs.

For the floating-gate amplifier presented in Sec. 3.6, the simulation model was tuned to match the actual data taken from the circuit. This post-fabrication tuning was achieved by minimizing the error between the simulated data and the actual data in MATLAB. MATLAB was used to control the SPICE simulation of the floating-gate amplifier, and was allowed to tune the values of the coupling capacitance, oxide thickness, junction depth, tunneling voltage, and source voltage. MATLAB was then instructed to tune these parameters until it minimized the error, using the simplex search method presented in [69]. For our example application, we were able to minimize this error with less than 5% change to the voltages and less than 1% change to the device properties.

3.6 Comparison of Model to Fabricated Designs

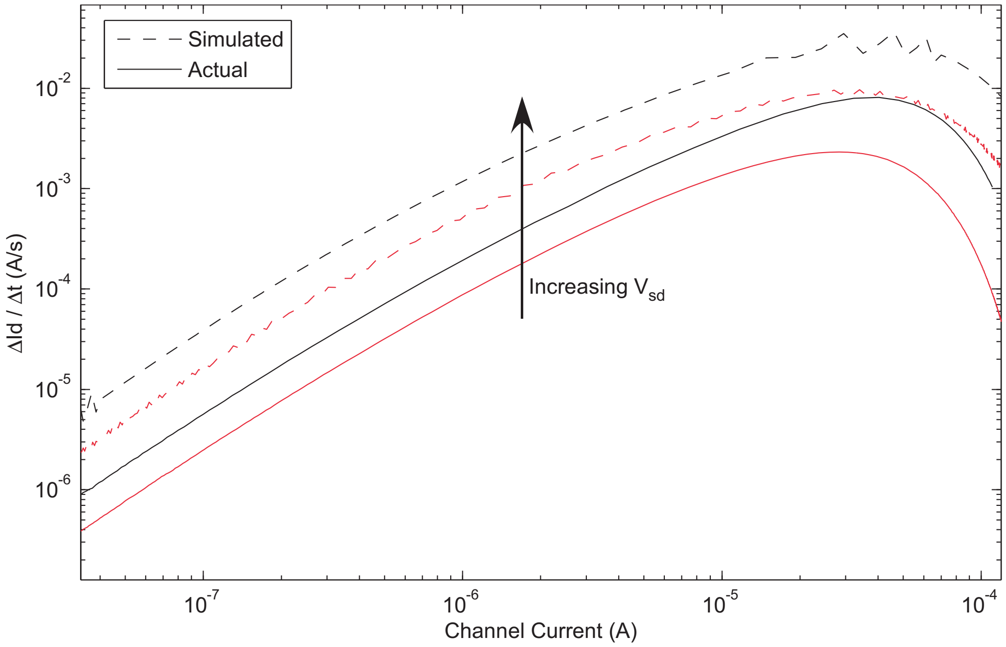


Figure 3.8: Simulated and measured data from a pulse-based injection programming scheme. The source to drain voltages used were 7.5 and 8 volts.

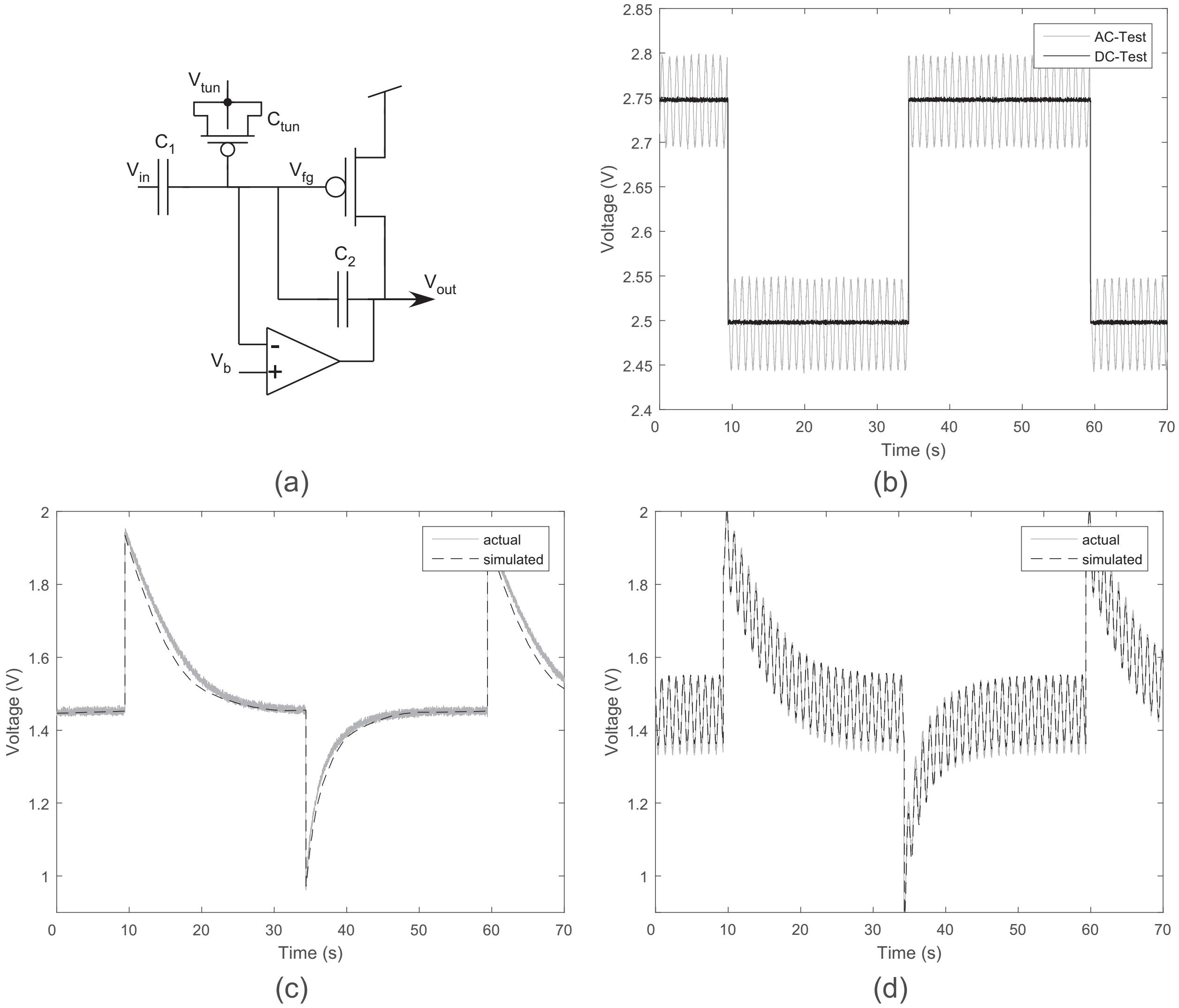


Figure 3.9: (a)The amplifier autozeroes to the level set by V_b through the constant use of injection and tunneling. The resultant waveform is an amplified version of the input, with the DC component tuned out. (b) The inputs used in our tests of the FG amplifier. (c - D) The fabricated and simulated results of the DC and AC tests. These demonstrate good agreement between the injection and tunneling currents of the fabricated and simulated device.

In this section we will demonstrate both the accuracy as well as the versatility of our macro-model. Data simulated within a SPICE-compatible environment will be compared to data taken from circuits fabricated in a 0.5μm process.

3.6.1 Static Programming

A common means of affecting the charge on a floating-gate device is pulsed-based injection programming. In this method, a device is allowed to inject for a short duration of time, the state of the channel current is then checked, and the process is repeated. This process is complicated by the non-linear change in channel current for injection pulses as seen in Fig. 3.8.

Figure 3.8 also demonstrates our models agreement with fabricated results. These simulation results were gathered using only device data made available from the foundry. Closer agreement could be reached with post-fabrication tuning.

3.6.2 Dynamic Programming

A unique use of a FG device is in a circuit which induces constant tunneling and injection, such as the circuit [70] shown in Fig. 3.9. The AC and DC simulations of this circuit demonstrate simultaneously the ability of this macromodel to accurately reproduce capacitive coupling, injection, and tunneling effects to create an auto-zeroing effect within an amplifier. Despite the fact that the tunneling and injection currents were on the orders of fempto-amps, the accurate simulation of these circuits made their implementation relatively simple.

3.7 Conclusion

We have presented in this chapter a floating-gate macromodel which is simultaneously capable of modeling capacitive coupling as well as charge programming effects. We have also presented parameters which are effective in the modeling of hot-electron injection as well as Fowler-Nordheim Tunneling across different fabrication sizes. In addition to this, we have presented some ‘best practices’ and insights into the layout of these devices. Finally, we have demonstrated the effectiveness of this model in a dynamically programmed environment.

Chapter 4

Extrema Sampling – An Adaptive Sampling Method

Extrema sampling, otherwise known as max/min sampling or peak sampling, is an adaptive sampling scheme which adjusts its rate to the changing frequency content of the target signal. In this chapter, it is tested against other continuous-time adaptive-sampling methods as well as traditional Nyquist Sampling. Unlike more complicated adaptive-sampling methods, extrema sampling relies on a simple mathematical manipulation – finding the zero crossing of the first derivative of a signal. By sampling the signal in amplitude, and possibly time, during these derivative zero crossings, the sampling method naturally adapts itself to the changing frequency content of a signal. In addition to presenting and testing this sampling method, point-wise reconstruction methods are presented. Simulation results are presented.

4.1 Energy-Constrained Sampling

Whether the application is bio-signal monitoring with wearable devices, voice-activity detection, or data transmission among wireless-signal-network nodes, modern electronic applications demand high data-fidelity with as low a power budget as possible. The means which are used to sample and preprocess (usually compress) a signal are a major part of this power budget.

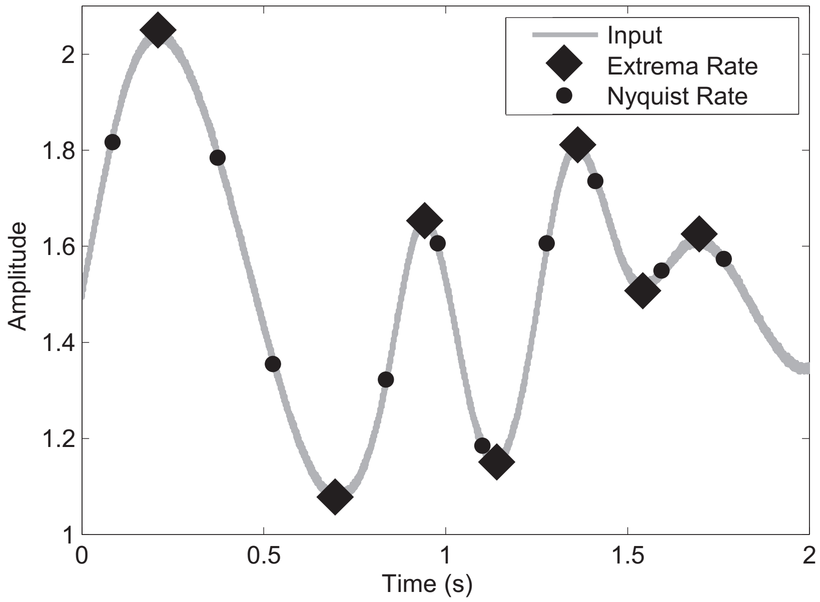


Figure 4.1: Comparison of constant-rate Nyquist sampling versus an adaptive-sampling method, extrema sampling. The Nyquist sampling rate is dependent upon the highest frequency content of a signal. In extrema sampling, only the local maximums and minimums are sampled. In this way, extrema sampling adapts itself to the changing frequency content of a signal.

The most direct method toward reducing the power expended on sampling a signal is to reduce the number of samples taken. Unfortunately, traditional Shannon-Nyquist sampling sets the sampling rate at a hard minimum of twice the highest frequency content of a signal. While the Nyquist rate has the advantage of being very well understood and characterized, it allows for power reduction only through digital scaling and radical changes in quantizer topologies.

Adaptive-sampling methods change their rate of sampling dependent upon the signal itself. Certain adaptive-sampling methodologies translate to physical, circuit-level, quantizer topologies more easily than others.

Level-crossing is the most popular of these physically translatable adaptive-sampling methodologies. In level-crossing analog-to-digital converters (LC-ADC), a signal is sampled at the instants where it crosses some pre-defined ordinate thresholds. In this way, the signal does not become oversampled during periods of relative inactivity or during periods of complete DC settling. The drawback to this method, however, is that periods of particularly 'bursty' activity may become oversampled. Consider the dominant spike in a QRS complex of an electrocardiogram signal, for example. This spike would undoubtedly traverse many threshold levels. But if the spike is of relatively constant derivative, then how much extra information is gleaned between the first and last threshold crossings? In this manner, it is possible for level-crossing sampling to over-sample a signal.

A less well known adaptive-sampling method is extrema sampling. Extrema sampling, also known as peak sampling or min/max sampling, takes samples only at the local minimums and maximums of a signal (Fig. 4.1). The advantage of this method is that it adapts itself to the changing frequency content of a signal, oversampling only if the usual condition of bandlimited is not met. The obvious and most unattractive feature of this method is that it requires computation of the derivative of a signal. However, low-power implementations have been demonstrated in fabricated systems which only attempt to find an accurate zero crossing of the derivative [24, 71]. This advantage cannot be understated, as it is know that taking a full and accurate derivative is an extremely costly task [72].

This chapter will quantify the advantages of adaptive sampling, with a focus on extrema sampling. In Sec. 4.2, extrema sampling will be further explained within the context of adaptive sampling. Sec. 4.3 will then explore methods of reconstructing extrema-sampled signals. We will then compare the sample reduction and signal fidelity of these different reconstruction methods with signals sampled using level-crossing and derivative-level crossing sampling in Sec. 4.4. Finally, we will then discuss hardware implementation in Sec. 4.5 before concluding in Sec. 4.6.

4.2 Adaptive Sampling

The crux of adaptive sampling is taking a signal and applying some mathematical manipulation to it to determine an adjusted rate of sampling. This differs from traditional Nyquist-rate sampling where the rate of sampling is twice the highest frequency contained in the target signal, regardless of what portion of the signal is currently being quantized.

One important caveat that simplifies the communication problem further down the chain is that, from the perspective of the receiver, an adaptive sampling method does not have to be asynchronous (4.2). The transmitter could still broadcast signals at the Nyquist frequency

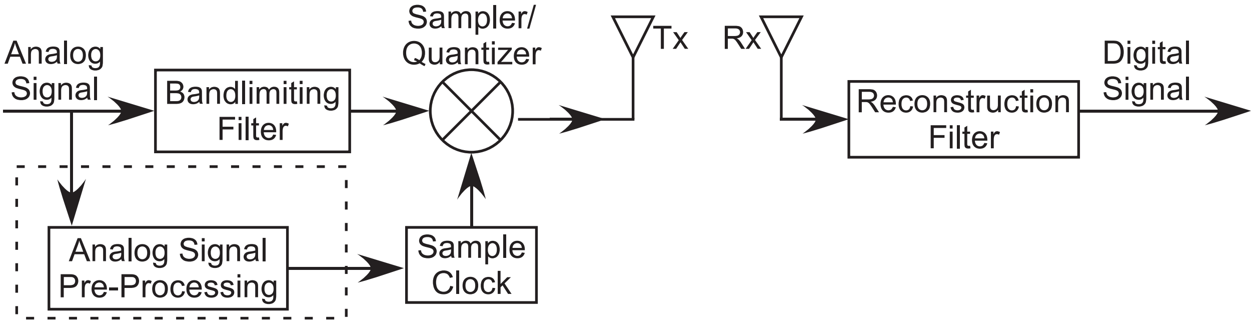


Figure 4.2: Block diagram of a signal quantization, transmission, reception, and reconstruction system. The dotted portion represents some form of analog pre-processing (e.g., level-crossing, extrema detection, etc.) and is required for asynchronous or adaptive sampling.

and still find power savings by transmitting a null state when the adaptive method has not found an updated sample. The assumption here is that the null state will be transmitted with a minimal codeword, thus preserving the majority of the power savings at the transmission and reception stages that a sub-Nyquist sampling rate would garner.

Performing transmission at the Nyquist rate, regardless of updated samples, has the added bonus of alleviating the need for discretized time-sample transmission. If the adaptive samples are not generated faster than the Nyquist Rate, the error introduced by assuming that the sample occurred at the time of transmission is limited to the period defined by the Nyquist Rate – thus restricting the error to within some reasonable bounds.

4.2.1 Extrema Sampling

Extrema sampling is a newer area of interest within the scope of adaptive sampling. This method identifies the local maximums and minimums of a signal, resulting in samples which adapt to the changing frequency content of the signal.

One drawback which is important to note about extrema sampling, and is true for any sampling scheme which is inherently asynchronous, is that you are no longer sampling only the ordinate of the system. Instead, both the ordinate and the abscissa, which is typically time, must be sampled. This effectively doubles, assuming equal resolution in both domains, the length of the sample's generated code word. Not only does this time quantization waste energy in the form of codeword transmission and generation, but it also wastes energy by requiring its actual sampling. In other words, it creates the need for another quantizer.

The energy expenditure of having to quantize in the time domain could be partially mitigated as described at the beginning of this section. By forcing the transmitter to broadcast at the Nyquist frequency, an extrema sampler could still operate at sub-Nyquist frequencies. The sampler would then update the transmitter when it has arrived at a new extrema value. If the transmitter has not received an update during a Nyquist defined window, then it could simply broadcast a minimal codeword to describe a null state.

4.3 Reconstruction

Reconstruction from irregularly spaced samples has been studied and proven possible in many works [73]. However, it stands to reason that if an adaptive method is being utilized for a particular application, then the resources available are quite constrained. Therefore, less computationally intensive methods of reconstruction may be desired. These methods also prove useful for when the sample number is limited – a case in which traditional reconstruction methods falter.

The two most common point-wise reconstruction methods are zero-order hold and linear interpolation. In zero-order hold, a sample is held until the next sample occurs. The resulting waveform appears like a staircase, with steps spanning from sample to sample. Linear interpolation is simply creating a line from point to point. Linear interpolation tends to yield more accurate results, but zero-order hold has the advantage of being non-causal, so signals may be interpreted in real time as they are sampled.

Another reconstruction method which is fairly simple to implement computationally is the method of Bézier curve fitting. 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 extrema values. Thus, we take N extrema 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 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

$$\begin{aligned} V(x) =&(1-x)^3 V_k + 3(1-x)^2 x V_k \\ &+ 3(1-x)x^2 V_{k-1} + x^3 V_{k-1} \end{aligned} \tag{4.1}$$

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

$$\begin{aligned} T(x) =&(1-x)^3 T_k + 3(1-x)^2 x T_{kmid} \\ &+ 3(1-x)x^2 T_{kmid} + x^3 T_{k-1} \end{aligned} \tag{4.2}$$

The result is a vector of time values that shift the sinusoidal curve to the appropriate time endpoints. The results of this reconstruction have been compared to a more traditional, and complex, windowing function and have been found to have similar results [74].

4.4 Results

When making the comparison between various adaptive-sampling-rate methods, it is best to consider them in the context of some class of signal. In this section, we present simulation results of extrema sampling for three different types of reconstruction: zero-order hold (ZOH), linear interpolation, and Bézier curve fitting. We will compare these to the results of level crossing sampling (LCS) and derivative level crossing sampling (DLCS) reported in [75].

In this comparison, we are utilizing the mean square error (MSE) normalized for constant amplitude, DC offset, and delay error. Since these errors are essentially a constant-scaling issue within the amplitude or time domain, they are easily corrected for in post-processing. We then calculate the signal error rate (SER) as

$$SER = \sqrt{\frac{\overline{x(t)^2}}{MSE}} \tag{4.3}$$

we will use this definition of SER, along with the sample rate, and the signal frequency to calculate the figure of merrit (FOM) defined in [75]:

$$FOM = \frac{N_s}{SER \times f} \tag{4.4}$$

The results in Table 4.1 indicate a clear savings in terms of the number of samples taken when the method of extrema sampling is used. On average, extrema sampling reduces the number of samples by over 75% when compared to LCS and DLCS. This reduction in sampling would translate to a power savings in the quantizer device (whatever form the ADC may take) as well as in the transmitter and receiver for networked devices. There is, however, a reduction in SER of 36-47%. Giving equal weight to the SER and the number of samples, as the defined FOM does, the reduction in samples outweighs the reduction in SER.

4.5 Sampler and Quantizer Implementation

It is prudent to make a choice in sampling paradigm based upon quanitative results, such as those presented in this work, but it would be inadvisable to do so without consideration of what circuitry will be required to implement a given paradigm. It is important to consider both the potential sources of error which can be introduced as well as the overall energy and space overhead created by the mathematical manipulations required for adaptive sampling.

First we will consider the most general implementation of a level-crossing system. Recall that this scheme works by identifying the times at which the signal crosses predefined thresholds. For the threshold definition, most techniques familiar to flash ADC architectures are acceptable, including the use of comparators with defined voltage references. This implementation still requires the use of either time quantization or a transmission rate, as mentioned previously, that is at the Nyquist Rate.

An offspring of level-crossing sampling is derivative-level-crossing sampling. Here, the first derivative is applied to the same threshold-crossing scheme that the original signal would normally be applied to in traditional level-crossing sampling. This method has the incredible advantage of being able to perform linear interpolation in real time, but it also has some practical drawbacks resulting from having to perform the derivative itself. Taking the derivative introduces a new source of noise and error, a new source of power consumption, and is also difficult to do over a wide range of frequencies [72].

Extrema sampling also requires the implementation of some form of a derivative, however this requirement is slightly relaxed. The only part of the derivative that need be accurate is its zero-crossing. While the derivative could be implemented with any standard derivative circuit [72], we have found that the use of an asymmetric envelope detector [76] works well over a reasonable range of frequencies, such as those required for voice detection [24] or bio-signal monitoring [71].

4.6 Conclusion

In this chapter, we have presented a fair comparison of asynchronous sampling methods. We have qualitatively explained extrema, level-crossing, and derivative-level-crossing methods, and have detailed their place within a quantizing and transmitting system, such as a wireless sensor network. We have then provided analysis of the performance of these sampling methods on a range of signal classes. Finally we concluded with some practical implementation considerations.

Table 4.1: Adaptive Sampling Rate Method Comparison

Signal	**System**	**SER(dB)**	**Ns (S/s)**	**FOM**
Single Tone 100Hz	LCS	37.6	12400	1.63
	DLCS	21.3	400	0.34
	Extrema - ZOH	7.1855	201.0051	0.279737
	Extrema - Linear	17.9894	201.0051	0.1117
	Extrema - Bezier	24.4036	201.0051	0.0824
Single Tone 3.9kHz	LCS	37.6	483600	1.63
	DLCS	66.5	483600	0.06
	Extrema - ZOH	7.2182	78010	0.2771
	Extrema - Linear	17.9909	78010	0.1112
	Extrema - Bezier	24.3962	78010	0.082
Dual Tone 200Hz and 2kHz	LCS	34.7	126400	2316/f
	DLCS	42.7	63200	463.1/f
	Extrema - ZOH	8.4746	4001.1	472.1249/f
	Extrema - Linear	17.3777	4001.1	230.2417/f
	Extrema - Bezier	25.3398	4001.1	157.8971/f
4kHz-bandlimited Random Gaussian (HPF:100Hz)	LCS	29.1	86000	3016/f
	DLCS	38.2	125100	1535/f
	Extrema - ZOH	2.3636	269.95	114.211/f
	Extrema - Linear	11.0617	269.95	24.4040/f
	Extrema - Bezier	10.8962	269.95	24.7748/f
ECG (HPF:0.5Hz)	LCS	27	139.3	6.22/f
	DLCS	29.1	166.1	5.79/f
	Extrema - ZOH	13.5595	130.0614	9.5919/f
	Extrema - Linear	19.3135	130.0614	6.7342/f
	Extrema - Bezier	22.2794	130.0614	5.8378/f
Speech (HPF:300Hz)	LCS	23.1	11930	835/f
	DLCS	27	15530	693.7/f
	Extrema - ZOH	3.7269	79854	2142.7/f
	Extrema - Linear	13.8299	79854	577.3979/f
	Extrema - Bezier	12.7326	79854	627.1559/f

Chapter 5

Temperature Compensation of Floating-Gate Transistor Arrays

Analog pre-processing has been shown to be energy efficient in a wide variety of low-power applications. Reprogrammable analog devices have leveraged this energy efficiency and applied it to a wider application space, but they still require accurate and stable bias currents for proper operation. For single-application devices, it is sufficient to create temperature compensation schemes that apply to a single bias current. But for reprogrammable or reconfigurable platforms which can be used in a variety of applications, temperature compensation must work well over a large range of potential bias currents and for a large number of different components. This chapter presents a temperature compensation method for floating-gate transistors in a reconfigurable system which improves performance over a wide range of currents and temperatures.

5.1 Introduction

Analog computation and pre-processing has been used in a wide variety of systems to improve energy savings, showing in some cases the equivalent of a 20-year leap in digital scaling [77]. Traditional analog pre-processing stages tend to be highly specialized application-specific systems, but developments in reconfigurable field-programmable analog arrays (FPAAs) [26, 78] have allowed these analog techniques to be applied to systems without *a priori* knowledge of the application space. One of the biggest hurdles in implementing reconfigurable analog systems lies in the infrastructure of the system. Temperature compensation is a particular challenge since the diverse application space demands a wide range of stable bias currents.

Many reconfigurable analog systems utilize floating-gate (FG) transistors to provide programmable bias currents [26, 78]. Unfortunately, the programmable bias currents generated by FG transistors are quite sensitive to temperature. There have been some successes in implementing temperature compensation for FGs employing large passive devices; however, these techniques are too area-hungry to be a viable option in dense FG arrays [79]. Others have employed a varactor on the FG node and use an additional voltage to modulate the capacitance at the FG node in response to temperature effects [80, 81]. This varactor-based method has been implemented on-chip and off-chip with great success, but has only been demonstrated for a temperature range between 25-43$°C$ due to the small tuning range of the varactor. Moreover, no work has yet demonstrated the ability of a temperature compensation circuit to accurately regulate a multitude of currents across an array of floating-gates, as would be utilized by an FPAA system, without adding unfeasible levels of power or area overhead.

This work explores temperature compensation of FG transistors when the required number and values of currents remain unknown at design time. All plots depict measured results from an integrated circuit fabricated in a standard 0.35μm CMOS process.

5.2 Floating-Gate Devices

FG devices are most commonly implemented as flash memory in digital systems, but they also have a memory-like application in analog systems. Within the context of analog systems, floating-gate devices can be programmed to hold a specific amount of charge on the gate, which is electrically isolated by a coupling capacitor. By programming specific amounts of charge on the gate, and thus programming the channel current to a specific value, the FG transistor becomes a tunable current source.

FPAAs often utilize large arrays of these FG devices [26, 78]. Within the FPAA system in this work, there are over 300 biases realized by FGs which control elements ranging in granularity from current-starved inverters to bandpass filters. This wide range of elements necessitates a vary wide range of bias currents – creating a need for a temperature compensation circuit which can stabilize a wide range of currents.

Before operating an FG device as a current source, it must first be programmed. There are two common programming mechanisms for modifying charge on an FG: Fowler-Nordheim (FN) tunneling and hot-electron injection. FN tunneling is typically used as a global erasure for all FGs since it is difficult to tunnel individual FGs. The procedure for FN tunneling is accomplished by significantly increasing the tunneling node capacitor (referred to as V_{tun} in Fig. 5.2). Under these conditions, charge is drawn off the FG node. Hot-electron injection is typically used to add electrons to the FG. This is accomplished by raising the FG transistor's V_{DD} to generate drain current conditions favorable for impact ionization. The 'hot electrons' with enough energy to surmount the FG barrier contribute electrons to the FG.

A transistor's operational characteristics will have an inherent dependence on temperature. Furthermore, transistors operating in the sub-threshold region, which is our main application operation area, experience more extreme changes (exponential) in channel current for a change in temperature than when in above-threshold operation.

Figure 5.1 shows the extent of temperature effects on an FG transistor. This example demonstrates for a single programmed FG device with a fixed V_{cg} – the voltage node for setting the target current bias – that the output current wildly varies with temperature change. Holding V_{cg} constant is the traditional method for setting target bias currents in FPAAs; however, a compensation circuit to modify V_{cg} in response to a change in temperature is clearly needed.

To generate temperature compensation, we use an FG current multiplier, as illustrated in Fig. 5.2 [81]. Considering that M_{REF} and M_1 have the same W/L, the charge stored on their respective FGs can be modified and used to ratio the reference current I_{REF} to I_{M1}. Operating an FG in the sub-threshold saturation region can be characterized by the following:

$$I_d = I_o e^{-\kappa V_{fg} q/kT} e^{V_s q/kT} e^{V_d/V_A} \tag{5.1}$$

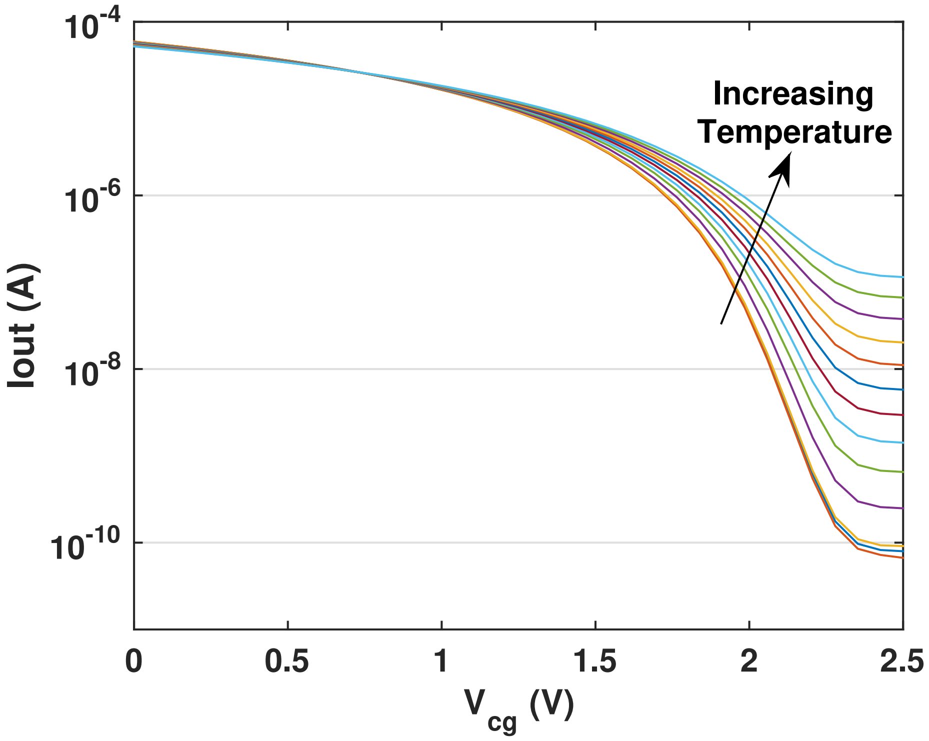


Figure 5.1: Temperature dependence of an FG transistor. The plateaued currents for $V_{cg} > 2.25V$ are artifacts of many junction connections to a single global connection where the current reading was taken. As a consequence, their collective leakage current becomes non-negligible and is manifested in these measurements. The current in a single FG transistor continues below these values.

where all voltages are referenced to the well potential, V_{fg} is the FG voltage, V_A is the Early voltage, and $q\kappa/kT$ defines the subthreshold current slope with kT/q being the thermal voltage (k = Boltzmann constant). V_{fg} can be approximated as:

$$V_{fg} = \frac{Q_{FG}}{C_T} + \frac{C_{cg}}{C_T}V_{cg} + \sum \frac{C_{par}}{C_T}V_x \approx \frac{Q_{FG}}{C_T} + \frac{C_{cg}}{C_T}V_{cg} \tag{5.2}$$

where Q_{FG} is the amount of charge on the FG, C_T is the total capacitance seen at the FG node including parasitic capacitances, C_{cg} is the control gate capacitance, and C_{par} are the parasitic capacitances with V_x being the terminal voltages coupled to V_{fg}. The right-hand side is a reasonable approximation given that C_{cg} represents the majority of total capacitance C_T. Then, incorporating (5.1) and (5.2) into the FG current multiplier topology of Fig. 5.2 renders the following output current relationship:

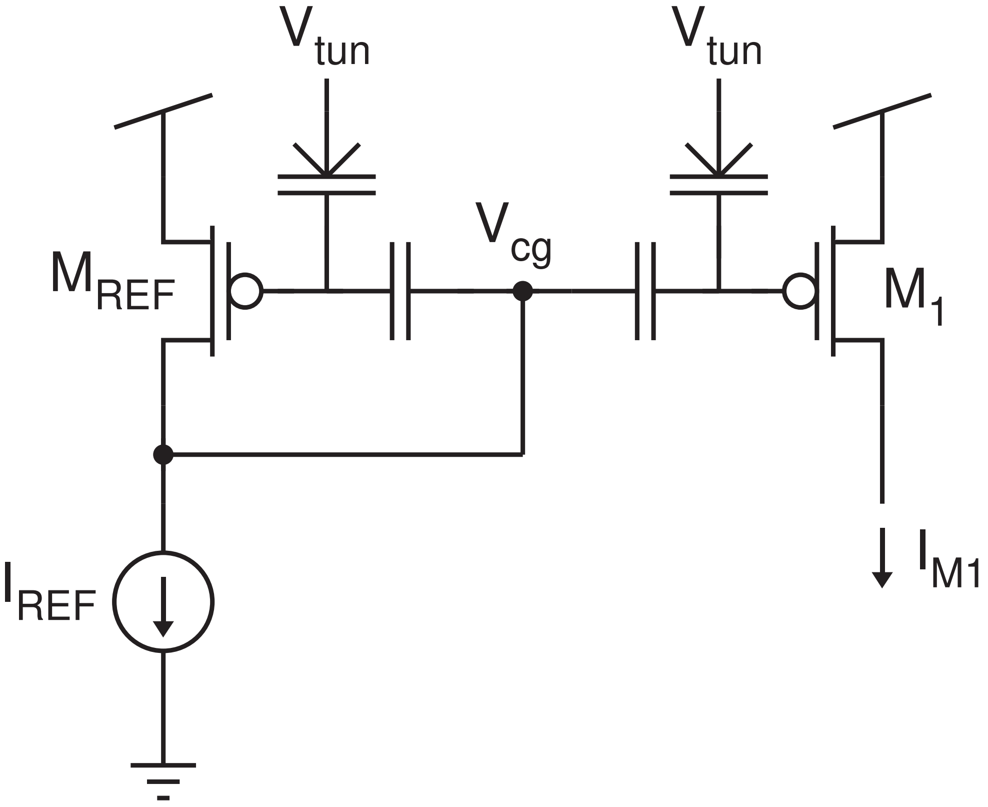


Figure 5.2: Floating-gate current mirror.

$$I_{M_1} \approx I_{REF} exp \frac{q\kappa(Q_{M_{REF}} - Q_{M_1})}{C_T kT} \tag{5.3}$$

where $Q_{M_{REF}}$ and Q_{M_1} signifies the amount of charge on their respective FGs. Equation (5.3) shows that all temperature dependence is removed from the output current for charge-matched FGs in the mirror topology. For FGs with unmatched charges in the mirror, there still exists a temperature dependence, but its effects are greatly diminished compared to an FG without temperature compensation. Unmatched charges have a temperature dependence that can be characterized for the following two cases: $Q_{M_1} < Q_{M_{REF}}$ and $Q_{M_1} > Q_{M_{REF}}$, where a larger charge amount is the result of fewer electrons on the FG and will correspond to a smaller current. Defining the exponential terms in (5.3) as β with the exception of T

$$\beta = \frac{q\kappa(Q_{M_{REF}} - Q_{M_1})}{kC_T} \tag{5.4}$$

gives the following expression of I_{M_1} for the two differing charge cases:

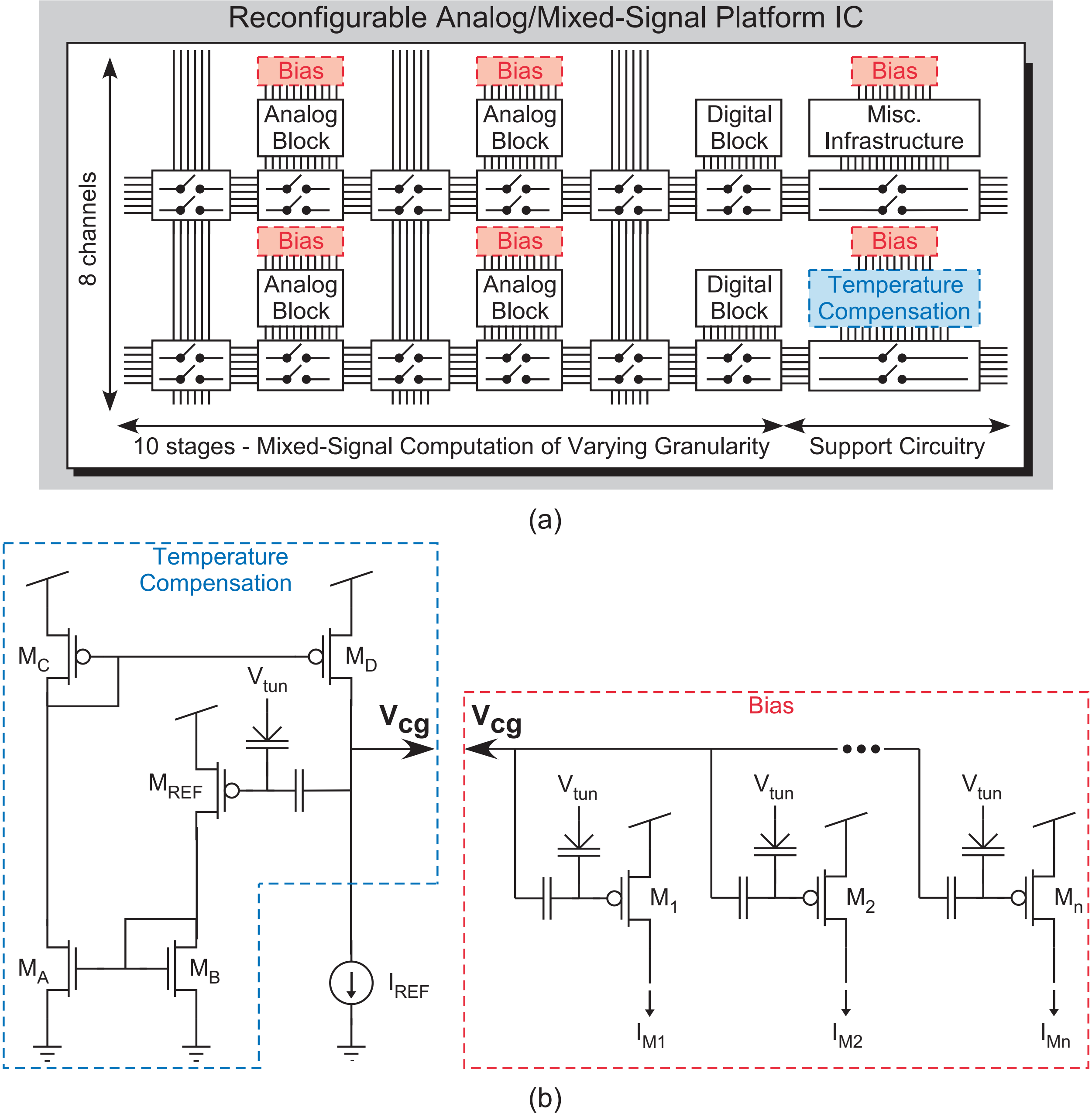


Figure 5.3: (a) FPAA block diagram showing the position of the specialized CAB which houses the FG temperature compensation structure. (b) Floating-gate temperature compensation structure showing connection to global V_{cg}. Currents $I_{M1} - I_{Mn}$ source to an nFET current mirror before connecting to the CAB circuits.

$$I_{M1} = \begin{cases} I_{REF}e^{\beta/T} & Q_{M_1} < Q_{M_{REF}} \Rightarrow I_{M_1} > I_{REF} \\ I_{REF}e^{-\beta/T} & Q_{M_1} > Q_{M_{REF}} \Rightarrow I_{M_1} < I_{REF} \end{cases} \tag{5.5}$$

Taking the derivative of (5.5) with respect to temperature yields a negative temperature relationship for $I_{M_1} > I_{_{REF}}$ and a positive relationship for $I_{M_1} < I_{_{REF}}$

$$\frac{dI_{M1}}{dT} = \begin{cases} -\frac{I_{REF}\beta}{T^2}e^{\beta/T} & Q_{M_1} < Q_{M_{REF}} \\ \frac{I_{REF}\beta}{T^2}e^{-\beta/T} & Q_{M_1} > Q_{M_{REF}} \end{cases} \tag{5.6}$$

These temperature coefficients will be manifested in the current measurement slope over a temperature range and become more apparent with larger differences in charge.

5.3 FG Temperature Compensation

The application presented in this chapter is applied to our FPAA, which is called the Reconfigurable Analog and Mixed-signal Platform (RAMP) [26]. To provide temperature compensation to such a large-scale system, we are leveraging the FG current mirror shown in Fig. 5.2. As stated in Section II regarding FPAAs, the RAMP utilizes floating-gates to provide precise, but temperature-dependent current sources. The FG current mirror is used to generate a control gate (V_{cg}) voltage that responds to changes in temperature and reduces its effect on current variation.

5.3.1 System Architecture

Our RAMP includes over 300 controllable current sources generated from FGs to be used as biases for the different circuits included within the RAMP. As an FPAA, the RAMP utilizes “Computational Analog Blocks,” or CABs, as building blocks for post-fabrication reconfiguration. The CABs can include simple devices or full circuits. By routing the CABs together and making connections to the FG biases, analog and mixed-signal systems can be synthesized directly on the RAMP. Figure 5.3 shows a block-level diagram of the RAMP and how the bias currents, as well as how temperature compensation fits in to the system as a whole.

To implement the FG current mirror for temperature compensation, a “reference” transistor is set-up in diode connection to dynamically set the global V_{cg} so that a steady current will be seen on any other FG connected to the mirror topology. To accomplish this, a specific CAB has been added to the RAMP that allows for the diode connection. Figure 5.3

shows the full schematic of the compensation circuit within a floating-gate array. Transistor M_{REF} is placed in diode connection via the current mirrors comprised of transistors M_A-M_D. Transistor M_D mimics the behavior of the M_{REF}, specifically at its drain. All reference transistors are sized identically to ensure the same current flowing from the drain of M_{REF} is also flowing from the drain of M_D. The drain of M_D is then connected to the global V_{cg} node, allowing for the complete diode connection of M_{REF}.

This topology employing two current mirrors out of the reference FG transistor is used instead of a simple diode connection (i.e. drain connected to the control gate) to ensure that the drain of M_{REF} is kept at a relatively fixed potential. With a conventional diode connection, any fluctuations at the drain of the reference transistor would be parasitically coupled to the floating-gate as indicated by (5.2), causing potentially large fluctuations in channel current. By using the current mirrors to create the diode connection, the voltage seen at the drain of the reference transistor will be more constant. A similar method is employed with the FGs used as current references. Instead of connecting the FG directly to a circuit as a bias, a single nFET-based current-mirror is used to ensure that any fluctuations in the circuit will not affect the FG output.

5.3.2 System Programming

The first crucial step in programming our temperature compensation system is to determine a value of I_{REF}. I_{REF} is the stable, temperature-independent, reference current which the rest of the system will refer to as the temperature of the environment fluctuates. The closer I_{REF} is to the individual mirrored current values, the more accurately the system will be able to compensate. For this demonstration within the RAMP system, we chose a value typical of low-power analog bias currents – $10nA$. Generally, this choice should be made by matching I_{REF} to the average value of the expected currents of M_1-M_n or the current which is most sensitive to temperature that is being implemented in the design.

For a given I_{REF}, when M_{REF} is programmed and diode connected, a particular V_{cg} will occur. With a known value of V_{cg}, we can characterize the programming of M_1-M_n. The programming is controlled by a continuous-time feedback circuit, similar to the one presented in [82]. The continuous-time programmer injects the FG to some value dependent upon a user-specified target voltage. At a given V_{cg}, this FG value will generate some specific current in M_n which can be stored in a look-up table to relate the value of the programmer's target voltage to the resultant current in M_n for a given V_{cg}. We can then use this look-up table to program the currents of M_1-M_n to any specified value.

A comparative view on the effectiveness of temperature compensation is demonstrated in Fig. 5.4. This shows the temperature compensation performance relative to the room temperature programming target from -25 to 85°C. Each line in Fig. 5.4(a) corresponds to a different ratio between the current I_{REF} and the current flowing through transistor M_n (depicted in Fig. 5.3(b)). The current I_{Mn} was programmed at room temperature (25°C) and adjusted via the FG temperature compensation structure shown in Fig. 5.3.

The large variance in current ratios shown in Fig. 5.4 is a constraint imposed by the nature of the RAMP, allowing for the device to span a wide range of applications without limiting the range of available bias currents. The best case is when the current targets between the FG (M_n) and FG reference (M_{REF}) are equal. As predicted by (5.3), ratios other than $1:1$ will result in less temperature compensation. With these ratios, which are a result of differing FG charges, a positive (negative) trend versus increasing temperature is the result of a negative (positive) difference in the numerator of (5.3). However, despite not working as well as a $1:1$ ratio, these cases still perform better than an umcompensated scenario, as shown in Fig. 5.4(c).

5.4 System Performance

Figure 5.1 shows the exponential dependence of temperature effects on uncompensated FGs. Due to the nature of the RAMP, target currents used for the operation of specific, synthesized circuits will be unknown until the end-user picks a desired application. Subsequently, depending on the complexity of the synthesized design, there will be more than one target current, at more than one target value. This calls for the ability to apply temperature compensation for a wide range of FG injection targets from a single targeted I_{REF}.

To show the advantage of temperature compensation in the RAMP system, a current-

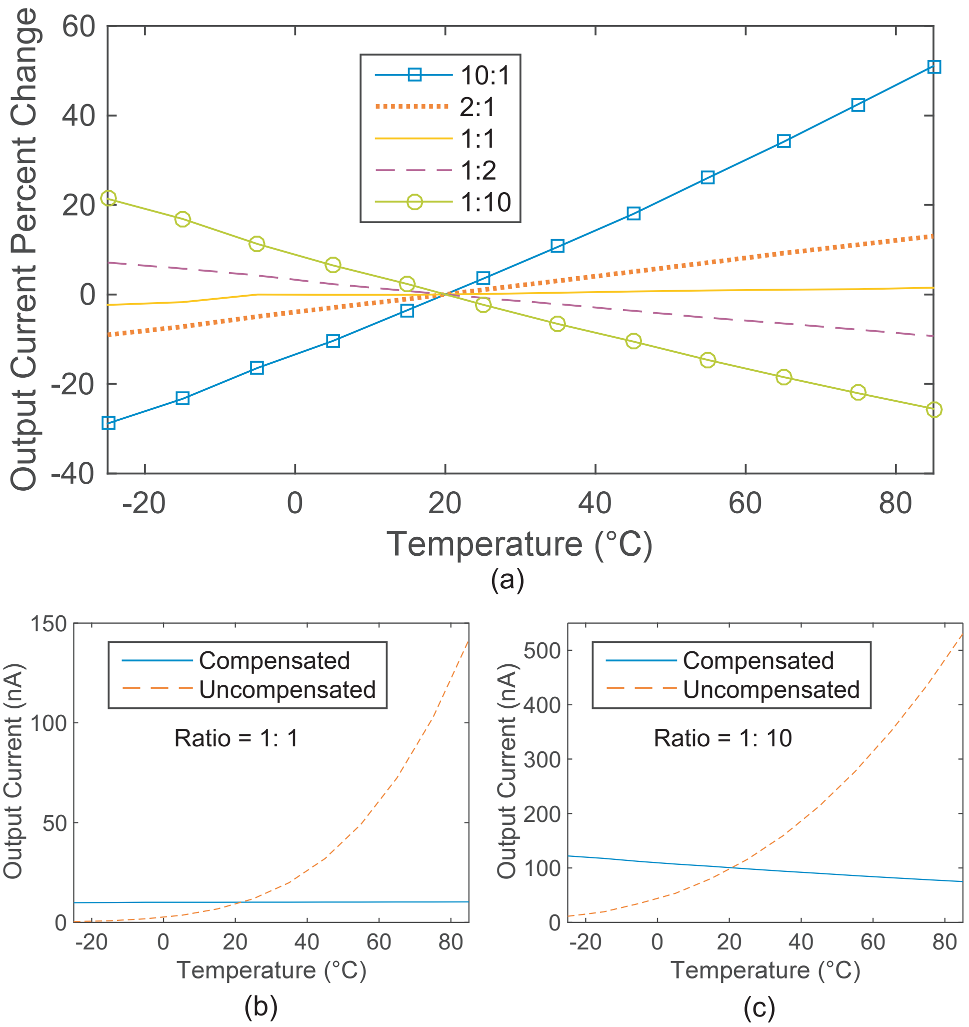


Figure 5.4: (a) Percent change in output current of an FG with temperature compensation normalized to room temperature. (b) and (c) show the effects of compensation compared to an uncompensated case for a ratio of 1:1 and 1:10 respectively.

controlled ring oscillator has been synthesized from one of the CABs. This circuit utilizes an input current, I_{in} (generated by one of the FGs) to starve its odd number of inverters, producing output frequency oscillations proportional to the input current (Fig. 5.5). For an output frequency of 10kHz, I_{in} was set to $33nA$. The I_{REF} current chosen for the temperature compensation system was set at $20nA$. Utilizing a ratio of $1 : 1.65$ for I_{REF} and I_{in}, the compensation scheme is able to decrease the fluctuations of the bias current over the temperature sweep of 0-90°C.

The ring oscillator was tested with both an uncompensated and temperature-compensated

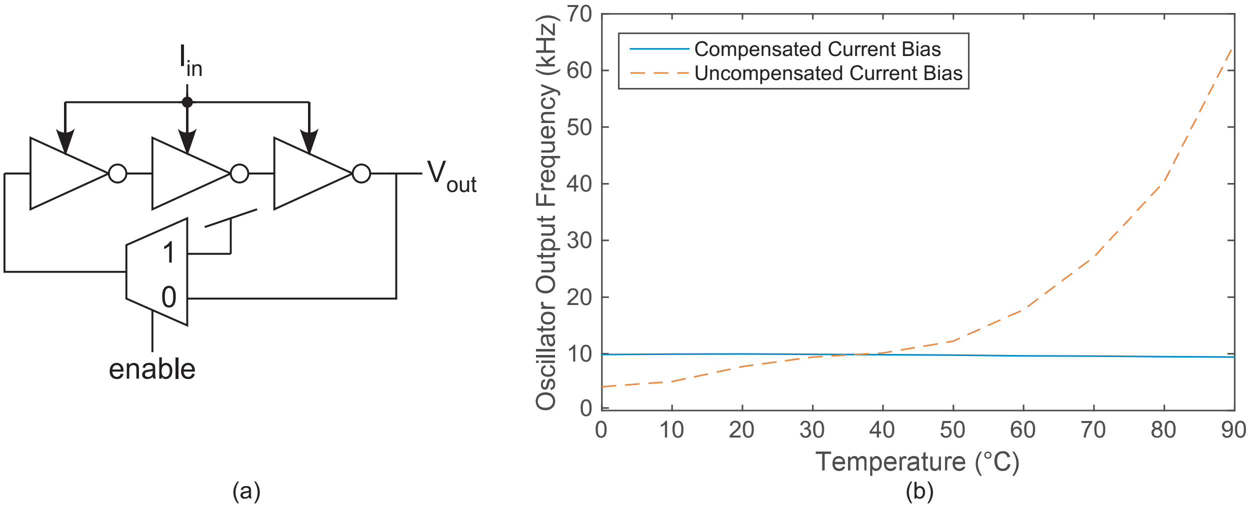


Figure 5.5: (a) Schematic of the ring oscillator synthesized in the RAMP circuit. (b) Ring oscillator frequency output with respect to temperature for a compensated and uncompensated FG current bias.

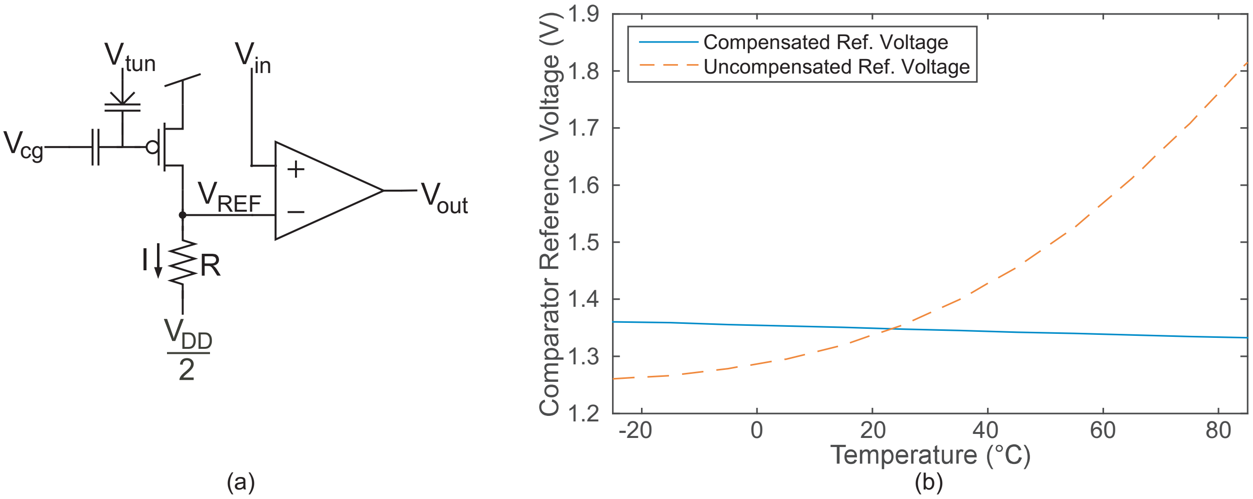


Figure 5.6: (a) Schematic of the comparator reference cell synthesized in the RAMP circuit. (b) Programmed comparator reference value with respect to temperature for a compensated and uncompensated FG current bias.

FG. The oscillator output frequency for both cases is shown in Fig. 5.5. The current bias using an uncompensated FG changes exponentially with increasing temperature while the compensated current bias remains close to the same value for the full temperature sweep.

A synthesized comparator example is shown in Fig. 5.6(a). The FG is programmed to draw a current across a resistor to set a desired reference voltage level, V_{REF}. The uncompensated and compensated V_{REF} measurements are shown in Fig. 5.6(b) where the reference current was set to $10nA$ and the current through the resistor was programmed to be $100nA$ for a ratio of $1:10$. The temperature compensation ratio does not represent an ideal case, but greatly outperforms the uncompensated case.

5.5 Conclusion

A temperature compensation circuit was presented for FG-dense structures such as an FPAA to improve performance over a temperature-varying environment. It is able to compensate for a multitude of FG biases with various output currents, which is representative of the variable nature of FPAA usage. The compensation system has been demonstrated to work with an array of FG current sources intended for biasing components such as an oscillator or other analog blocks. Its performance has been tested as a part of the larger FPAA system and shown to improve current bias stability.

Chapter 6

Asynchronous ADC with Reconfigurable Analog Pre-Processing

Many energy constrained devices such as cell-phones and wearable medical devices utilize sparse or bursty signals—signals characterized by relatively short periods of high activity. Traditional Nyquist-Rate converters are an inefficient tool for converting sparse signals, as a great deal of energy is wasted converting portions of the signal with relatively-low information content. Asynchronous sampling methods attempt to circumvent this inefficiency by sampling based upon the characteristics of the signal itself. However, many asynchronous sampling solutions struggle with a robustness/resolution trade-off. In this chapter, we present a new paradigm in which a flexible analog front-end is paired with an asynchronous successive approximation data converter and an asynchronous time-to-digital converter. The result of this paradigm is that the system can be adapted to individual applications, allowing specific data points to be targeted and avoiding data conversion inefficiency. The system will be demonstrated along with the example applications of measuring an electromyography waveform, a vocal waveform, and the QRS-complex within an electrocardiogram waveform. The demonstrated system, fabricated in standard 0.5μm and 0.35μm processes, produces minimal voltage/time pairs while consuming 5.96μW of static power.

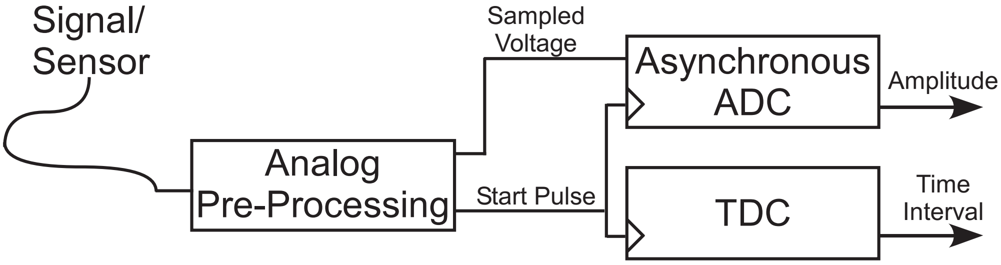


Figure 6.1: Asynchronous analog-to-digital conversion system. A reconfigurable analog front end reduces information to only the relevant data points and also triggers the subsequent blocks which then produce digital words for the corresponding voltages and time intervals.

6.1 Asynchronous Data Conversion

Passive voice monitoring in cell-phones, wearable bio-medical devices, and a variety of other modern systems bear a common burden—monitoring sparse or bursty signals in a resource constrained environment. Sparse signals undergo short periods of high activity among longer periods of relative inactivity. This sporadic data content makes traditional analog-to-digital conversion using Nyquist-Rate sampling inefficient because the conversion of unnecessary or uneventful data yields little information. Such data-conversion inefficiency greatly impacts the overall power-budget of the system and is a key hurdle in enabling usability and proliferation of such energy-constrained devices.

Some data converters attempt to vary their sampling rate based upon the spectral content of the target signal. One of the more popular data-driven designs is the level-crossing ADC (e.g. [83]). Level-crossing ADCs require a number of threshold levels proportional to the desired resolution, but as the number of levels is increased, the bandwidth of the signal becomes limited. The work in [50] addressed this bandwidth issue by adapting the level spacing to the changing frequency content of the signal. But, in addition to the increased digital overhead, there is still the possibility of taking superfluous samples which yield no new or useful data. A potentially improved method would be to use analog pre-processing to identify specific portions of the signal which contain relevant data, while disregarding the other portions of the signal.

In this chapter, we propose an asynchronous sampling system which can be tuned for a variety of applications. The sampling system produces two digital signals—one propor-

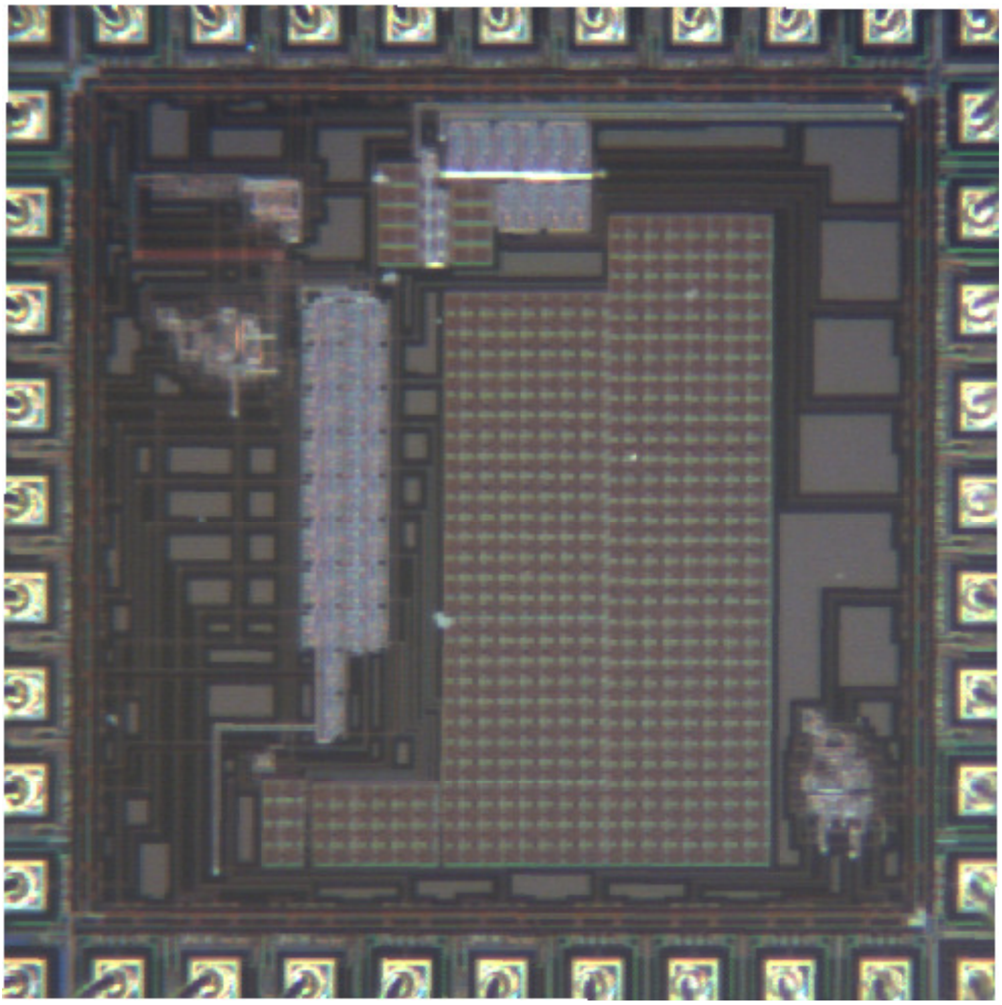


Figure 6.2: Die photograph of the asynchronous ADC/TDC fabricated in a 0.5μm standard CMOS process available through MOSIS.

tional to the sample amplitude and the other proportional to the inter-sample time period (Fig. 6.1). The trigger mechanism for these data-driven asynchronous measurements is controlled by an analog front-end implemented in our reconfigurable analog/mixed-signal platform (RAMP) system [26]. The use of a reconfigurable front-end allows us to create a sample triggering mechanism tuned to the minimal amount of required data. The well-tuned analog front-end combined with asynchronous data conversion allows for highly efficient data extraction.

The system presented here samples based upon the extrema-sampling paradigm which has been explored in [24, 74], but this system could be used with a variety of sampling methods, particularly for well understood and characterized signals. Section 6.2 establishes the overall system architecture which we used to implement this adaptive sampling technique. The individual components of the system are then examined in sections 6.3, 6.4, and 6.5. Finally, Section 6.6 concludes with an example usage of the system in an electrocardiogram (ECG) monitoring environment. Unless otherwise noted, all plots in this chapter are measured results from the ADC/TDC which was fabricated in a 0.5μm standard CMOS process (Fig. 6.2) or from the RAMP system [26] fabricated in a 0.35μm standard CMOS process all available through MOSIS.

6.2 System Overview

When sampling signals with sparse information, constant-rate sampling unavoidably creates extraneous samples during periods of low activity. Such inefficiency can be avoided by the use of analog pre-processing, which has been shown to be computationally efficient [22]. While a fully custom front-end would yield the highest energy savings, the use of the RAMP reconfigurable platform [26] allows us to adapt this analog front-end for a variety of applications. Many sampling paradigms could be used with this system, and we have chosen to demonstrate this system using extrema sampling.

The RAMP system identifies and samples the extrema values in a manner similar to that proposed in [24], locating the local minimums and maximums with two symmetric extrema detection and sampling circuits. For some applications, the extrema amplitude values and inter-sample time values are enough to extract useful information. By performing the identification of these values in the analog domain, it is possible to leave the data converter inactive until an appropriate value has been sampled.

Once an extrema value has been sampled, the ADC/TDC are signaled and begin their respective conversions (Fig. 6.3). The ADC utilizes a simple successive approximation architecture. The TDC is essentially a voltage-controlled oscillator with a digital counter attached to the output that counts the number of oscillations between extrema occurrences. By holding the voltage-bias constant and pausing/resetting the TDC after every sample, the total number of counted oscillations will be proportional to the elapsed time since the previous sample, thus enabling inter-sample time measurement.

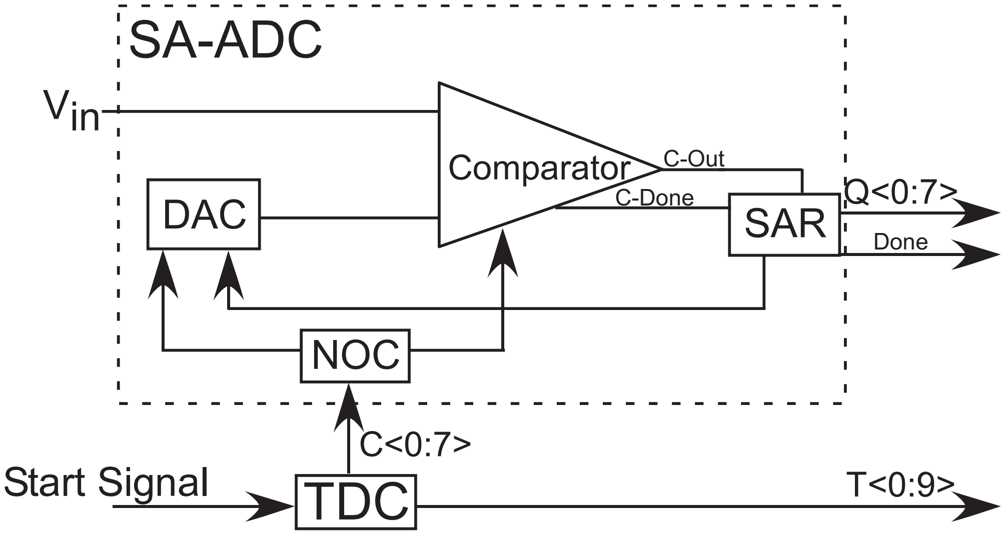


Figure 6.3: System diagram of the ADC/TDC showing the input, a held voltage and a start signal, as well as the system flow and outputs: bits representing the analog value, time elapsed, and a done signal.

6.3 Analog Front-End for Pre-Processing

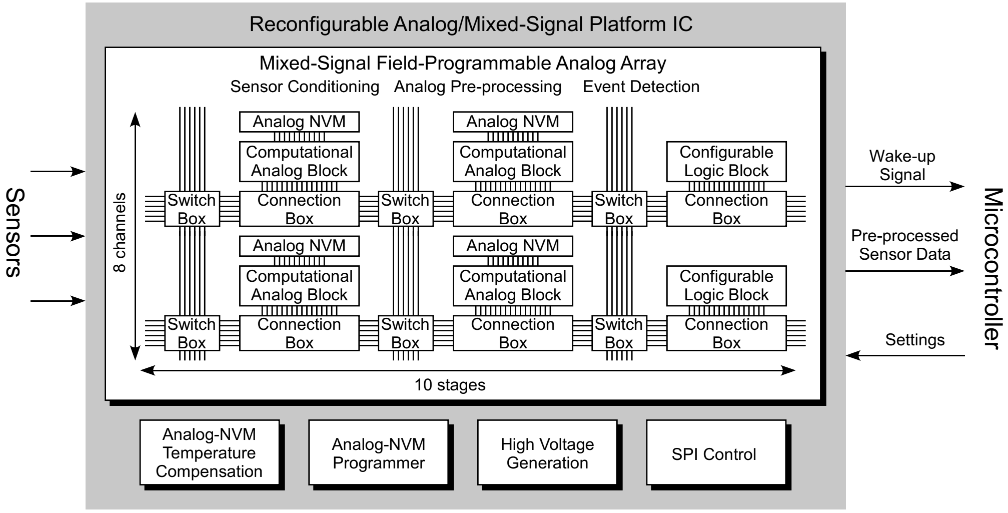


Figure 6.4: Architecture of the RAMP integrated circuit.

The RAMP is a field-programmable mixed-signal system which can be used to synthesize a variety of event-detection and signal-processing applications quickly and without the need for circuit-level expertise (Fig. 6.4). The signal-flow inspired architecture consists of 80 computational analog blocks (CABs) arranged across ten stages in eight channels. The stages provide design components of varying granularity grouped according to functionspectral analysis, transconductors, sensor interfacing, transistors, and mixed-signal operations. The eight channels enable the architecture to process signals in parallel. The RAMP provides further design flexibility through the usage of tunable circuit parameters (e.g., filter bandwidth). These parameters are realized through the usage of floating-gate transistors as a non-volatile analog memory [67].

We have also created a development environment to simplify the design of applications on the RAMP. This development environment includes (1) a custom netlisting language to describe connections on the RAMP, (2) an automated placement routine based on simulated annealing to choose the most appropriate circuit elements, (3) an automated routing algorithm that leverages a heuristic rules-based system for connecting the basic building blocks, and (4) an abstraction framework that allows hierarchical and reusable design. These individual tools together greatly simplify the design process for the end user and help to make the device selection and connection transparent to the user.

6.3.1 Synthesized Extrema Detection Circuit

Locating and sampling the extrema values of the signal was performed by an analog front end (Fig. 6.5) implemented within the reconfigurable analog RAMP system [26]. The RAMP system allows the user to specify different configurations of analog (filters, envelope detectors, multipliers, etc.) and digital components in a manner similar to operating an FPGA.

Locating the maximum extrema of the signal was performed by first taking the envelope of the signal. This envelope was set to track the input aggressively on the rising edges and to lag behind when the signal began to decline. The comparator then produced a logic high signal when the envelope lagged behind the input. This logic high value was used to trigger a pulse generator which provided the ‘event pulse’ signal. In addition to signaling the subsequent data converter stages, the pulse also triggered a sample-and-hold circuit which sampled the value of the input. All of these circuits were implemented on the RAMP. The sample was then provided to the data converter. The minimum extrema location was

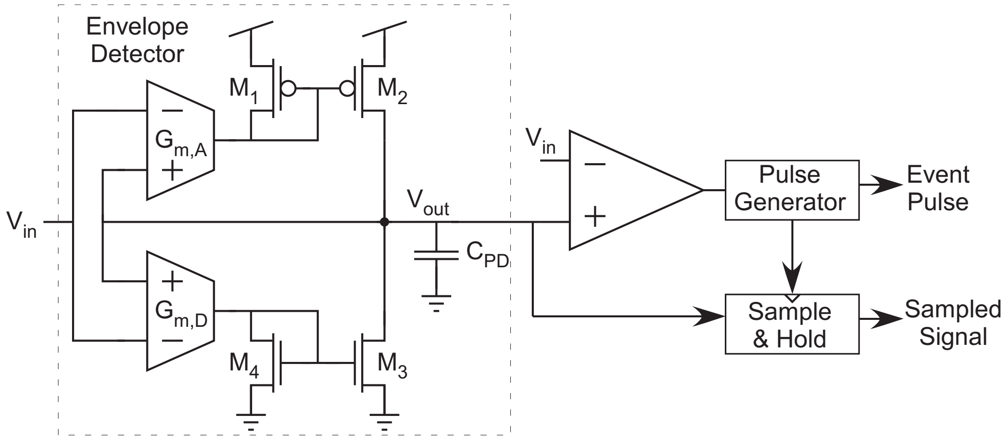


Figure 6.5: The maximum locator circuit which makes up one half of the analog front end synthesized in the RAMP system. The circuit consists of an envelope detector, a comparator, a pulse-generation circuit, and a sample-and-hold. The envelope detector tracks the input on the rising slope of the signal and then slowly decays on the falling slope of the input signal. This lagging decay is detected by the comparator—signaling a local maximum. The comparison signal is then converted into a pulse which triggers both the sample-and-hold circuit as well as the subsequent ADC/TDC stages. The minimum locator circuit is the symmetric equivalent, tracking the input on the falling slope and lagging the input on the rising slope.

performed with a symmetrically equivalent circuit.

6.4 Successive-Approximation ADC

The amplitude conversion of the extrema values was performed by a successive approximation ADC. The successive approximation architecture was chosen for its demonstrated power efficiency [20]. It also has the advantage of being appropriate for a relatively large range of frequencies and amplitudes, thus making it an appropriate choice for a variety of different systems with varying signal characteristics. This section will detail the various circuits used to create this ADC.

6.4.1 Successive Approximation Register

The successive approximation register (SAR) is the control circuitry used to run the binary-search-like process within an SA-ADC. The basic principle of this search is to test each bit from the MSB to the LSB and save the result. The boxed top half of Fig. 6.6

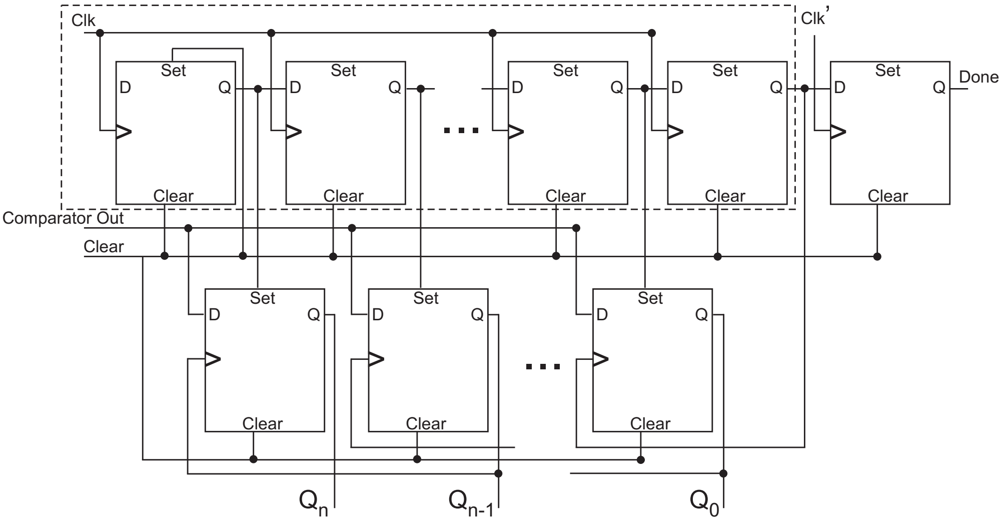


Figure 6.6: A schematic of the successive approximation register. The boxed area acts simply as a shift register. The lower set of D-flip flops is where the bits are actually applied to the rest of the system and stored appropriately.

is simply a shift register. This portion of the SAR passes a logic high down the chain of flip-flops upon each clock cycle. When a flip-flop in the shift register is set to logic high, a corresponding flip-flop below it is also set high. This bottom row of flip-flops applies the digital word to the DAC. When the next shift occurs, but before the DAC or comparator is updated, the previous bit is changed to reflect the output of the comparator. Logically, this is the equivalent of the flip-flop updating its output to reflect whether the guess was accurate or not. This process proceeds in a sequential manner until the last flip-flop outside of the dashed box on the top row receives the logic high. Reception of this state reflects that the SAR has finished all of its cycles, and thus the analog to digital conversion is complete.

6.4.2 Digital-to-Analog Coverter

For the SA-ADC to work properly, different reference voltages have to be generated which means some sort of DAC must be used. The charge scaling DAC (Fig. 6.7) is a popular choice for this application.

A charge scaling DAC works by taking the system reference voltage and dividing it in a binary fashion across a capacitor bank. By using binarily weighted capacitors, all reference voltages can be generated for a given resolution. The naive approach to doing this for an 8

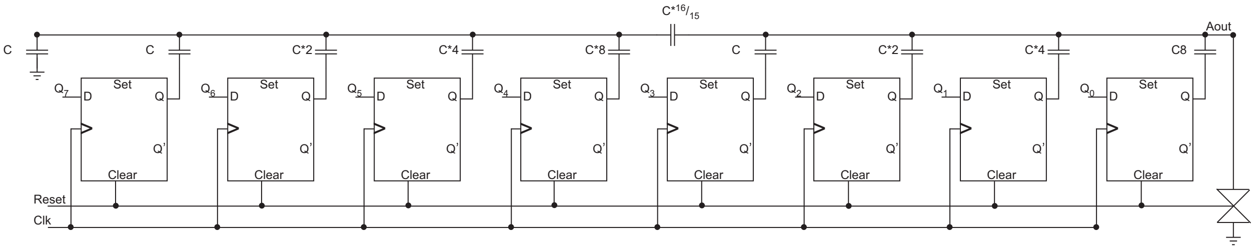


Figure 6.7: A charge-redistribution digital to analog converter (DAC). Binary-weighted capacitors allow for the generation of all necessary reference voltages for the given resolution. The bridging cap (pictured in a different orientation than the others) helps to reduce the component spread necessary to achieve higher resolutions.

bit system would be to have a bank of capacitors ranging from one unit capacitance all the way up to 128 times the unit capacitance. Doing so would create a very large component spread, which would effect the matching and overall performance of the system. A better solution is depicted in Fig. 6.7. Here, there are essentially two sub-DACs, each of four bits. The DAC to the left of the bridging capacitor acts as the LSB DAC and is scaled to combine appropriately with the MSB DAC.

The DAC of an SA-ADC is one of the more power hungry components as well as one of the more demanding to design well. Capacitors should be small enough that the power drain of the system is relatively low, while large enough to ensure that proper matching in the layout of the devices can be achieved. A charge-scaling DAC is overall a good choice for this architecture because it takes advantage of two of the things that CMOS fabrication does best: capacitors and switches.

6.4.3 Non-Overlapping clock

The non-overlapping clock generator (NOC) is a simple but important block in many digital subsystems, including this ADC. This ADC requires two phases of operation, which is to say that it requires two clock signals that are high at mutually exclusive time frames. The schematic diagram of this device can be seen in Fig. 6.8. In the top circuit, the clock pulse is logic ANDed with a version of itself delayed by two gate delays. The effect of this is that it is curtailed slightly on the leading edge during the two gate delays. The bottom circuit simply inverts the clock signal before performing the same operation. The net result is two

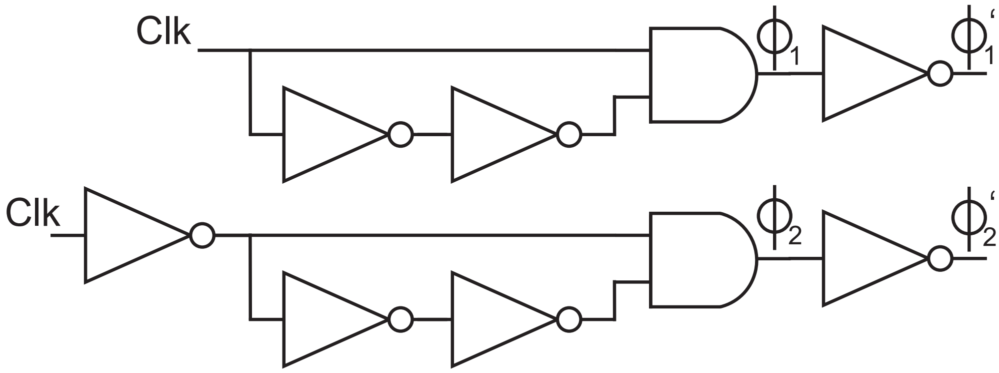


Figure 6.8: Schematic diagram of a non-overlapping clock generator (NOC). Takes a clock and outputs itself as well as itself in the opposite phase with a slight lag time between the two where both outputs are logic low.

clock signals that are out of phase and non overlapping by a gate delay on either side. More complicated designs with, for example, cross-coupling between stages have been developed which help eliminate the possibility of meta-stable points at high frequency operation, but ultimately they are unnecessary for this application.

6.4.4 Comparator

Maximizing resources within energy-constrained systems is a high priority, but it is difficult, for example, to convert voltages along the entire full-scale range. The comparator shown in Fig. 6.9 was based on the design in [84] and was modified to maximize system resources by enabling rail-to-rail conversion. Normally, a comparator is implemented with either a pFET-based differential pair, which suffers at higher voltages, or an nFET-based differential pair, which suffers at lower voltages. Ours uses both complimentary versions of the pFET-based and nFET-based comparators and selects which one to use based upon the result of the first successive approximation.

During the first successive approximation, it is already known that one of the comparator inputs will be at the mid-rail voltage. Therefore, either a pFET-based or nFET-based comparator would be able to perform the comparison. We chose the nFET-based version to perform the first conversion. The result of this conversion signifies whether the input is in the top-half or bottom-half of the full scale range. If it is found to be in the top-half,

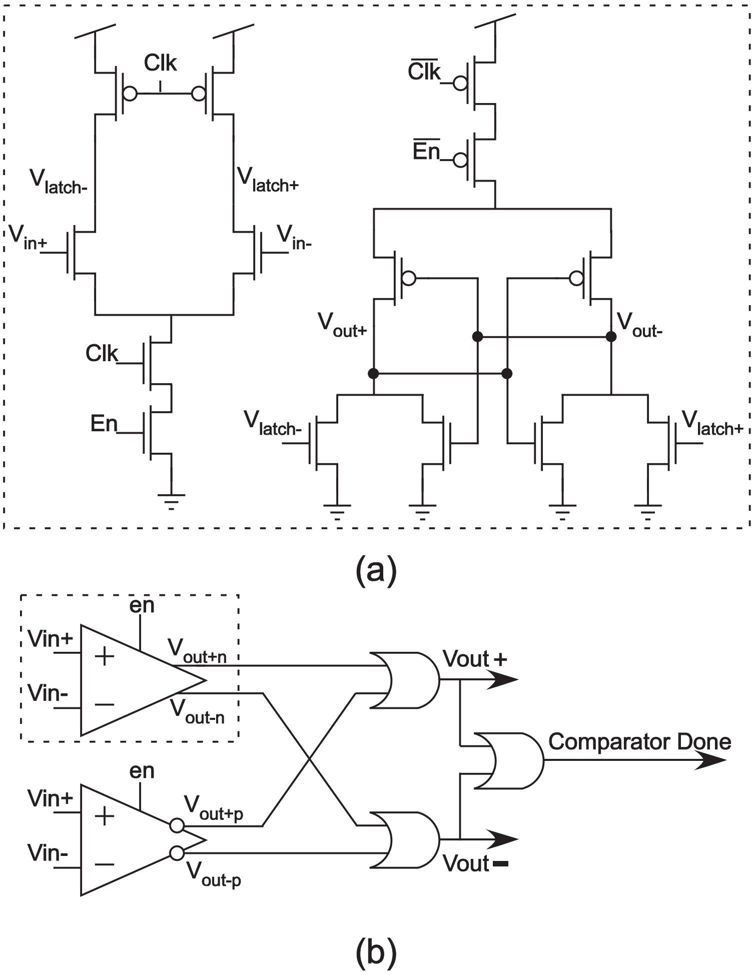


Figure 6.9: Comparator schematics. (a) The clocked comparator is designed to use negligible static energy and very little dynamic energy by utilizing minimally sized transistors. Prior to comparison, the clock signal is low, which pre-charges the V_{latch} nodes. When the clock goes high, the V_{latch} nodes are discharged at rates proportional to the V_{in+} and V_{in-}. These V_{latch} nodes control the subsequent voltage latch stage. The end result is that the output node that corresponds to the higher input voltage will be pulled high. (b) The nFET-comparator is paired with its pFET equivalent. The appropriate comparator is chosen by the MSB of the successive approximation register, thus ensuring that the comparator never operates in a region where it cannot make an accurate comparison. These outputs are then OR'd together to create a comparator done signal.

we perform the remaining conversions using the nFET-based comparator. Conversely, if the input is found to be in the bottom-half, the remainder of the conversion is performed using the pFET-based comparator. By intelligently selecting which comparator is used, we are able to convert values along the entire full-scale range.

Once the appropriate input pair is selected, the matched FETs conduct at a rate proportional to the input voltages applied to their gates. The nodes that are discharged by these FETs are attached to a clocked latch which cause one output node to become logic high while the other output node becomes logic low. The result signifies which voltage input was greater.

The drawback to using two symmetric comparators is that two different input-referred offsets must be accounted for in post-processing. For this implementation, no layout matching techniques were used to address this unequal offset, but it would be a viable technique for mitigating the issue. Another method would be to use programmable floating-gate devices in place of the input pairs. The floating gates could then be programmed with the appropriate charge so that the offsets between the two halves of the comparator matched [41].

6.5 Time-to-Digital Converter

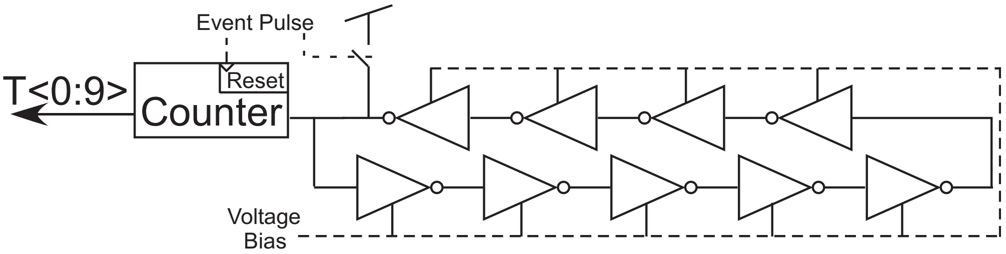


Figure 6.10: Time-to-digital converter consisting of a voltage-controlled oscillator and a counter. The counter keeps track of the number of oscillations. The resulting digital word is proportional to the time elapsed since the last event pulse. The event pulse clears the counter and also shorts part of the oscillator to logic high, which ensures that it begins in the same state after every reset.

The inter-sample times are recorded using a time-to-digital converter (TDC). A time-to-digital converter is simply a device which takes some periodic or known signal and uses it to estimate the time interval between signal pulses [85].

Our TDC implementation was designed to take a minimal amount of room and a minimal amount of digital support circuitry. The periodic signal was created by current-starved inverters arranged in a voltage-controlled oscillator topology (Fig. 6.10). The current starving (Fig. 6.11) ensured that we could both mitigate the power wasted by short-circuit current and that we could control the frequency of oscillation of the overall device. The voltage bias

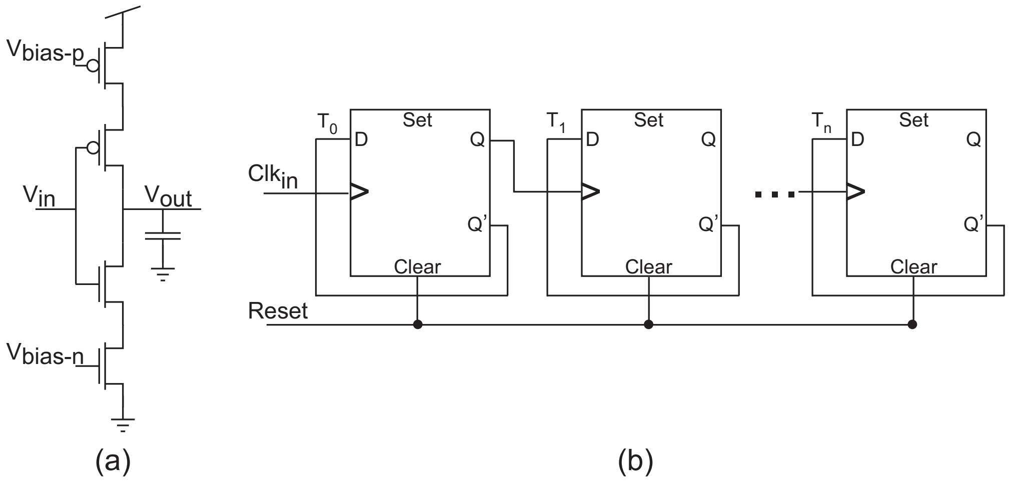


Figure 6.11: (a) A simple delay unit. The delay is proportional to the size of the capacitor and inversely proportional to the current which flows through the inverter. (b) A simple binary counter. The D flip-flops are sequentially linked and output successive binary numbers.

was held at a DC value and the output oscillations were recorded using a digital counter. The output of this counter was then a binary word which was proportional to the length of time since the last restart pulse. The restart pulse was provided by the RAMP system and was used to reset the counter values to zero when a new sample was detected.

6.6 System Example

The first task in implementing an asynchrnous peak sampling system is identifying and sampling the extrema values. By performing this pre-processing in the analog domain, we were able to detect peaks at a measured static power consumption of 4.95μW. When the analog front-end signals the occurrence of an extrema, the TDC halts operation, allowing readout of the digital word which represents the time duration since the previous extrema occurrence. The TDC is then reset, allowing it to begin counting up until the next extrema occurrence. The TDC can operate with frequencies ranging up to tens of kilohertz, but for these tests, the TDC was operated at a frequency of 1.15kHz for a measured power consumption of 1.01μW. It should be noted that this component will greatly benefit from smaller fabrication processes and lower power supplies.

The analog front-end also simultaneously signals the ADC to begin converting the newly sampled extrema. The ADC has an extremely low static current draw measured at 14.75nW. The low static-current means that the ADC can remain powered on between samples without worry of draining the power budget. The simulated energy per conversion was 47.4nJ.

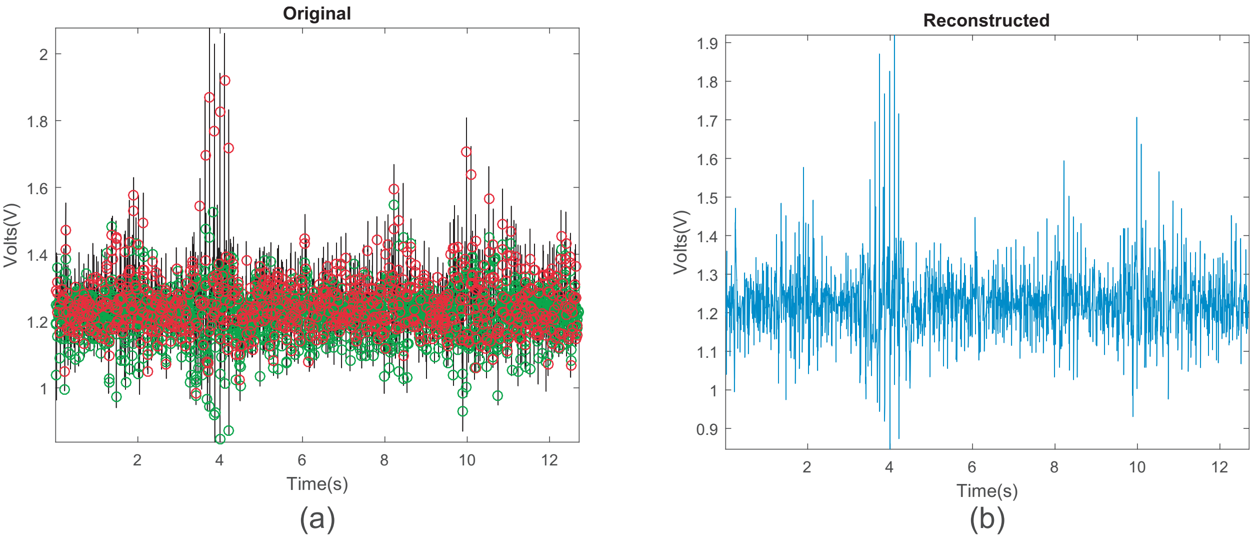


Figure 6.12: (a) The original EMG waveform with successfully detected peaks. (b) The recreated EMG waveform using Bezier reconstruction.

The first application demonstrated here is the capture and quantization of extrema values within an electromyography (EMG) signal (Fig. 6.12). EMG signals are traditionally used to diagnose muscle and nerve health. In addition to measuring neuro-muscular health, EMGs are gaining popularity in measuring muscular exertion in physical therapy and sport-science applications [86]. As these applications become more advanced, increasingly power and sample efficient devices will be required.

Figure 6.12 demonstrates the results of our presented system on an EMG waveform taken from the Example EMG database found at [87]. Over the 12.7 second signal, 3,035 peaks were successfully detected. Given that the highest frequency within the sample was about 2kHz, about 50,796 samples would be required for an equivalent Nyquist-rate sampling system. Therefore, our extrema sampling quantizer represents a greater than 16 times sample reduction over standard techniques, while maintaining a mean square error in reconstruction (using Bezier reconstruction) of less than 0.0036.

The next application involves the quantization of a speech recording. Low-power speech sampling and reconstruction is useful in IoT devices which actively listen for pre-defined

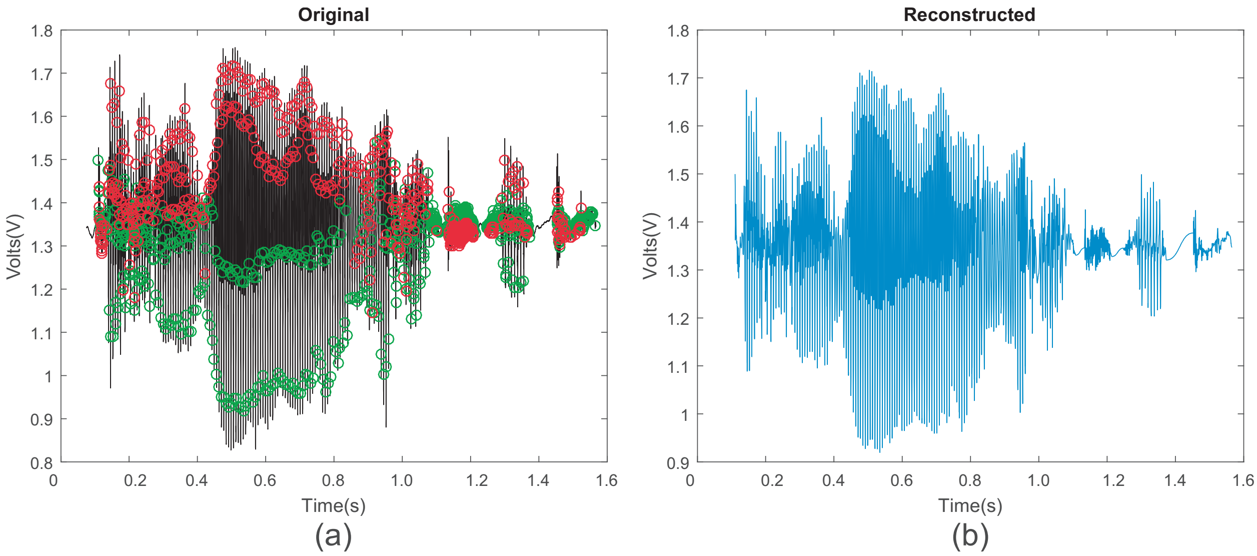


Figure 6.13: (a) The original vocal waveform with successfully detected peaks. (b) The recreated vocal waveform using Bezier reconstruction.

commands. In these applications, perfect reconstruction is not necessary. Instead, only a certain mean-square error threshold must be respected so the system can identify known commands.

Figure 6.13 shows the successful capture of 1,436 extrema values in a 1.45 second speech waveform. The fastest captured component of the signal is 5,333 Hz, thus dictating a minimum of 15,528 samples to be taken in a traditional Nyquist-rate sampling system. This nearly eleven times reduction in sample number maintains a 0.0483 mean-square error rate and represents a potentially-significant energy savings in a system which must process or transmit speech clips for command recognition.

Finally, we present an application where the goal is not reconstruction, but instead the capture of specific points of data. Electrocardiogram (ECG) waveforms are very well understood signals which have a limited number of medically relevant data points. Our example will focus on QRS-complex capture, disregarding the other portions of the ECG wave. QRS-complexes are useful for a variety of medical purposes ranging from monitoring for hyperkalemia or cardiac hypertrophy to simply extracting heart-beat to estimate perceived exertion [27]. For the purpose of demonstrating this system, an extrema sampling approach is a natural fit for reducing an ECG waveform to its QRS-complex—which can be viewed as

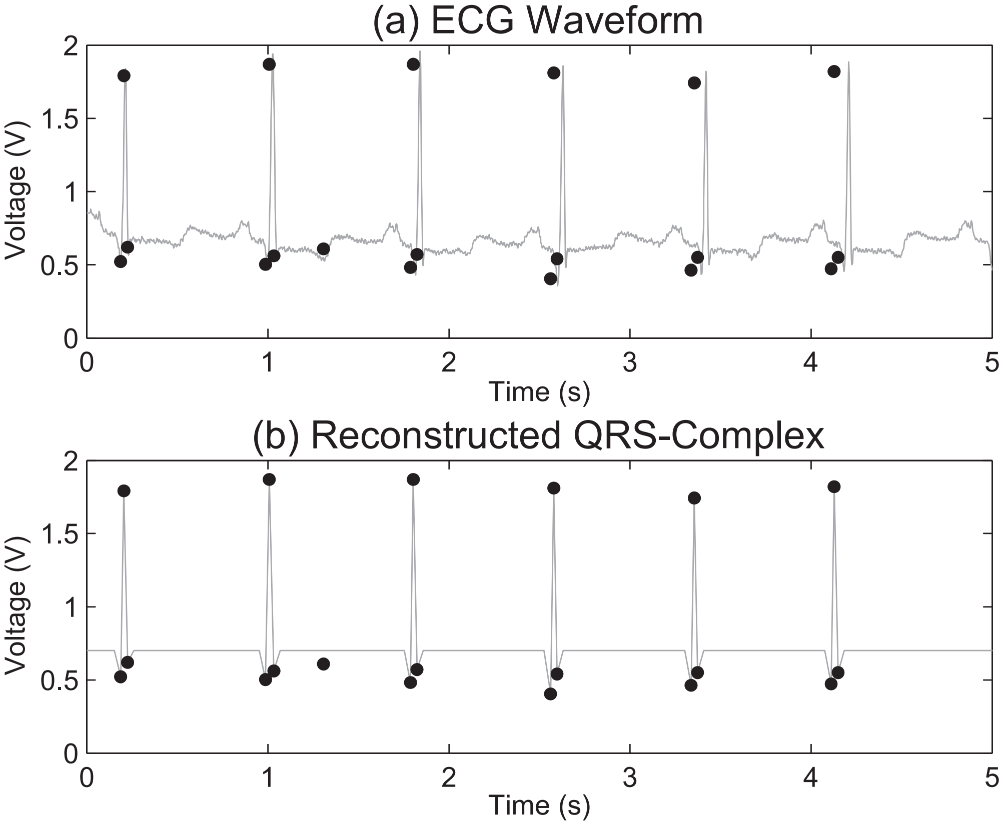


Figure 6.14: (a) The original ECG waveform (gray line), taken from the MIT arrhythmia database, with the detected extrema values (black). While the period-to-period temporal accuracy is relatively high, error aggregates creating the appearance of a shorter signal. This aggregate error could be addressed in post-processing. (b) QRS-complex reconstruction (gray) with the detected QRS values (black) as well as a single false positive.

a distinct local maximum between two local minima. By extracting only the QRS-complex and avoiding sampling extraneous data, this device would allow a wearable health or fitness system to be a more viable long-term option.

Figure 6.14 shows the operation of the system and the resultant approximations of the amplitudes and time intervals between the QRS peaks. The device was tested with real world data taken from the MIT arrhythmia database [87]

In viewing the operation of the entire system (Table 6.1), the advantages of adaptive sampling become readily apparent. If we define our FOM to be proportional to both the power consumed and the number of samples generated, the FOM will represent both the capture detection/quantization power consumption along with potential per-sample power consumption throughout the remainder of the system. In this regard, the demonstrated

Table 6.1: Performance Comparison

	This Work	**C.I. Ieong** [88]	**R. Abdallah** [89]	**H.M. Wang** [5]	**X. Zhang** [4]
Process (um)	0.35* /0.5**	0.35	0.045	0.18	0.13
Supply (V)	2.5* /3.3**	1.8	0.34	-	.3
Power (uW)	4.95* /1.01**	0.83*	0.33*	2.21*	0.034* /0.22**
Fs (Hz)	3-5	300	600000	500	1000
FOM (Fs*Power)	29.8	249	198000	1105	254
Method	Extrema Detection	Wavelet Transform	Stat. Error Comp.	Pan-Tompkins	R-Wave Triggered

*QRS Detector
**ADC

system yields the best performance. Figure 6.14 shows the system effectively capturing a QRS-complex. The highest frequency component, which is the change between Q and R, would determine the minimum sampling rate in a traditional Nyquist sampling scheme. For the demonstrated waveform, the 57Hz change would necessitate a sampling rate of 114Hz from a conventional Nyquist-rate ADC. This sampling rate would yield approximately 485 samples over the 4.228 seconds that the signal is demonstrated over. That amount of extra needless samples is greater than 25 times the number of samples taken in our system, even including the single false-positive.

6.7 Conclusion

We have presented a system capable of converting well-understood signals asynchronously and at a low-power cost. The samples were in the form of voltage/time pairs and allow valuable information to be taken from the signals with minimal energy expenditure. The system was demonstrated on real-world ECG signals and was shown to faithfully capture the QRS-complexes. We plan to extend this system past the extrema-sample paradigm to include other event-driven applications including voice-detection and gait detection.

Chapter 7

Extrema-Enhanced Successive Approximation ADC

Adaptive-sampling methods reduce power consumption in the quantizer and transmitter of discrete networked devices by lessening the number of required samples, but they burden the quantizer with the need for additional mathematical analysis. For example, extrema sampling has been shown to be advantageous in reducing sample numbers across different classes of signals, but it requires some additional power expenditure for finding the zero crossings of the first derivative. Ideally, this extra information would be used not only in the sampling rate and signal reconstruction schemes, but also in the actual sampling or quantizing of the signal. This work will propose a method for incorporating this extra information into the standard successive-approximation algorithm. The advantages of this method will be analyzed by mathematical simulation.

7.1 Data Converter Efficiency

Discrete signal monitoring devices such as wireless sensor networks [90], implantable biomedical devices [91], or wearable devices [71] provide extremely useful data and functionality, but must work within very energy-constrained environments. It is easy to only think about the sensors role in the energy budget, but these discretized devices must not only sense the phenomena, they must also quantize it, and often times they must also transmit it. The latter two operations often create a great deal of energy overhead [92].

The energy expenditure of traditional Nyquist-rate analog-to-digital converters (ADC) has been greatly reduced over the years, owing both to digital scaling as well as an incredible amount of research being performed on topological improvements [11]. However, Nyquist-rate ADCs are not always appropriate for energy constrained applications. Many of these applications monitor 'bursty' environments (i.e., applications characterized by long periods of inactivity interspersed with relatively brief events of interest). By sampling such signals at the Nyquist Rate, not only is energy wasted in quantizing superfluous samples, but energy is also wasted in transmitting them.

Adaptive-rate ADCs change their sampling rate based upon some characteristic of the target signal. By doing so, these ADCs are often capable of reaching sub-Nyquist rates. One example of a type of an adaptive-rate ADC is level-crossing ADCs (LC-ADC). LC-ADCs operate by sampling the signal only when it crosses predefined thresholds. While these systems are capable of achieving extremely low-sample rates and low-power expenditure [31, 50], they are still prone to oversampling in certain conditions. Specifically, during periods of constant derivative change, or spikes, it is possible for more samples to be generated than what is required for accurate reconstruction.

Extrema detection is another adaptive-rate method which provides sub-Nyquist rate sampling. By sampling only the local maximums and minimums of a signal, this method adapts itself to the changing frequency spectrum of a signal. Also, the very regular pattern generated by this method, maximums and minimums repeating in that order, lends itself easily to compression.

In this work, we will present an ADC system built upon the extrema sampling paradigm. This work will show not only the gains which can be made from sampling a signal with an adaptive rate, but it will also show how this sampling paradigm can be leveraged at the circuit level. We will begin in Sec. 7.2 with an overview of extrema sampling and how it can reduce the number of steps taken in a standard successive approximation ADC (SA-ADC). Sec. 7.3 will then detail the circuit-level implementations of this system. Finally, Sec. 7.4 will provide simulation results of this system before concluding in Sec. 7.5.

7.2 System Overview

Adaptive sampling rates can provide tremendous gains in energy savings just by the reduction in total samples alone, but ideally we could leverage the mathematical preprocessing to create more opportunities for energy savings in the quantization stage. Put another way, the analysis required by the sampler should be utilized by the quantizer. This section will explore how to leverage extrema sampling in the successive-approximation register of an SA-ADC. This discussion begins with a brief review of what the extrema sampling algorithm is, followed by recommendations in modifying the standard successive approximation algorithm.

7.2.1 Extrema Sampling

Extrema sampling, also known as peak sampling or max/min sampling, is a sampling methodology wherein only the local maximums and minimums of a band-limited signal are measured. Extrema detection can be achieved through the use of a zero-crossing detector attached to a derivative circuit or through an asynchronous envelope detector for added frequency range.

Regardless of the physical implementation, extrema locators have one important potential drawback as well as one important quality which can be leveraged by the quantizer. The drawback is that in the presence of high-frequency noise, it is possible for extrema sampling to generate many extraneous samples. It is therefore extremely important that the signal be properly bandlimited prior to reaching the extrema locator phase.

The advantage offered by extrema detection is that it implicitly adjusts the dynamic range of a sample based upon the previous sample. Put simply, a local minimum always follows a local maximum, and will always be less than that local maximum. Similarly, a local maximum always follows a local minimum, and will always be more than that local minimum.

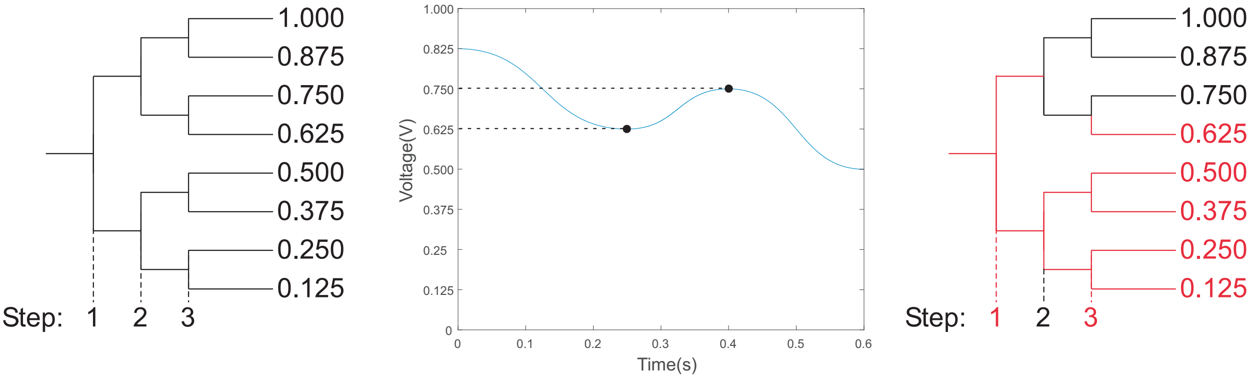


Figure 7.1: Representation of the dynamic-range-limiting capabilities inherent in extrema sampling. The far left graph represents a 3-bit SA-ADC and the eight possible quantization values which it may arrive at after 3 steps of successive approximation. The middle graph represents a local minimum sample preceding a local maximum. The far right graph represents the possible quantization levels and steps eliminated by the knowledge of the preceding value in quantizing the local maximum.

7.2.2 Extrema Successive Approximation

Before suggesting modifications to the standard successive approximation register of a SA-ADC, it is worthwhile to review its standard operation. A standard successive approximation register, essentially the finite-state-machine (FSM) of the SA-ADC, works similarly to a binary search. In quantizing a sample, it operates over some ordered list of possibilities – in the case of an ADC, this would be the range dictated by the full-scale-range of the system divided into increments determined by the resolution of the system. The result of this algorithm is that a sample is assigned a quantization level after a number of steps equal to the resolution of the system.

There are multiple ways to adjust the standard SAR to accommodate for extrema sampling, but the one presented here is meant to be compatible with existing SAR standard-cells. The advantage is that this can be implemented easily into existing designs. The disadvantage is that it does not speed up the conversion. It still takes the same number of clock cycles to finish, but it disables the digital-to-analog converter as well as the comparator during the unnecessary steps, thus leading to greater power savings.

Algorithmically, the FSM operates as such:

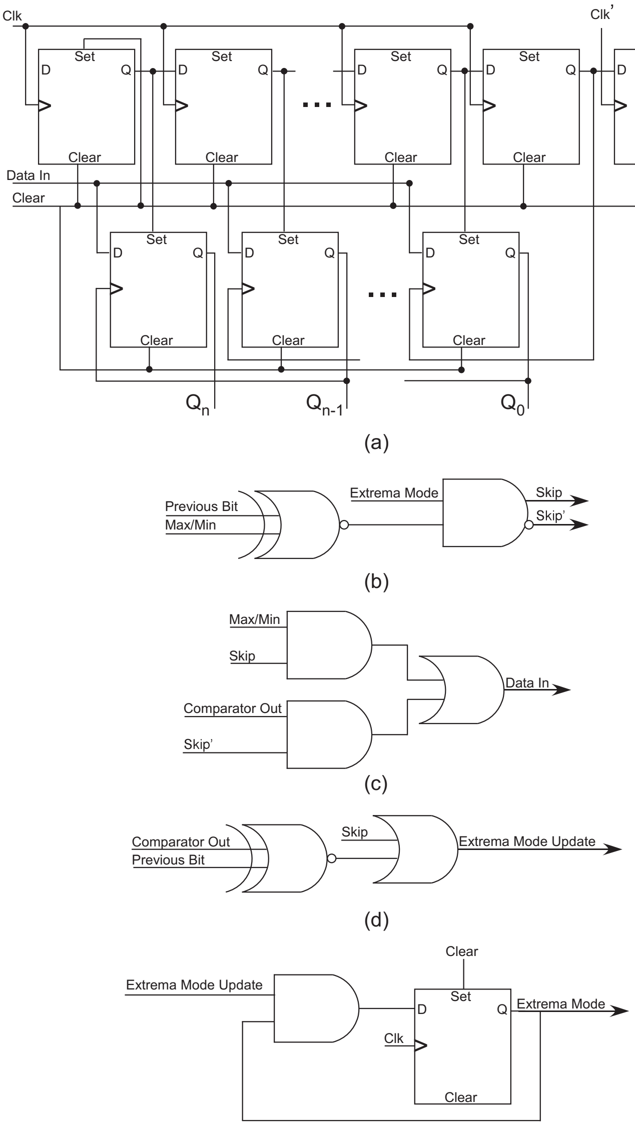
```
if lastVal==min
```

```
    mode=1
else
    mode=0
end

ExtremaMode=1

for number of bits N

    if ExtremaMode
        if previous_bit(n)==mode
            Disable DAC and Comparator
        else
            Enable DAC and Comparator
            evaluate current_bit(n)
            if current_bit(n)==mode
                ExtremaMode=0;
            end
        end
    else
        Normal SAR operation
    end
end
```

7.3 Subsystem Implementations

Implementation of the extrema-enhanced SA-ADC requires some modifications to typical SA-ADC circuitry. The most notable change is within the SAR, as detailed in Sec. 7.2. Another modification that we will propose here is to the comparator. A topological enhancement, along with the use of floating gates for analog memory, allows us to construct a full-scale-range comparator.

7.3.1 FSM Implementation



(a)

(b)

(c)

(d)

(e)

Figure 7.2: (a) A schematic diagram of a typical successive approximation register. The only modification to this portion is the Data In line, which replaces the typical direct line from the comparator. (b) This added stage of digital logic determines when it is appropriate to disable the DAC and comparator, effectively skipping a step in the conversion of a sample. (c) The digital logic here decides what value to place on the Data In line. It could create a logic high or low, if the current sample is being skipped, or it could pass the comparator output in the case of a normal SAR cycle. (d) In the case that a step is not skipped, but extrema mode has not yet been disabled, this phase determines if extrema mode should remain enabled. (e) The D flip-flop here maintains the current state of extrema mode, defaulting to on when a new cycle begins.

The structure is identical to that of an ordinary SAR, but with the addition of some logic gates as well as a D-Flip Flop (DFF). The DFF is set to output high at the start of the conversion and will remain high as long as the previous sample (in our example, a minimum) provides useful information about the current sample – we call this state extrema mode.

The logic is arranged such that it takes the current bits form the previous sample and uses them to dynamically limit the search of the SAR. This is done by looking at the previous value of the current bit being determined, and generating a 'skip' signal if it is a '1' in the case of a previous min, or a '0' in the case of a previous max. This 'skip' signal disables the DAC and comparator, possibly by blocking their clock input. It then loads the comparator output line with a '1' in the case of a previous min or a '0' in the case of a particular max.

If the 'Extrema Mode' signal is high, but the previous max/min does not have a 0/1 in the current bits place, then the word must be generated by the DAC, and the comparison must be made. If the result matches that of the previous max/min, then the 'Extrema Mode' signal remains high. If it does not match the previous sample, then the 'Extrema Mode' signal becomes low, and the rest of the word is converted using the standard SAR algorithm.

7.3.2 Comparator

The comparator is often considered one of the main design problems in ADC design as it it limits the resolution of the device and is typically one of the most power draining components. [19] and others have utilized the comparator shown in Fig. 7.3(a) as a high-resolution low-power solution. We can subrange a pFET implementation and an nFET implementation to create a full-scale range comparator by choosing the appropriate implementation based upon the determination of the first bit (Fig. 7.3(b)), as we demonstrated in [71].

Unfortunately, this subranging of comparators will cause a new issue which must be addressed: uneven input offset. It is expected in comparator design for their to be some input offset between the input pair, and usually this manifests as a slight shift in the transfer curve of the ADC. Unfortunately, we can expect an unequal amount of offset in between the pFET input pair and the nFET input pair. This unequal offset will cause a discontinuity in the ADC transfer curve. The solution is to use floating-gate elements in place of one of the

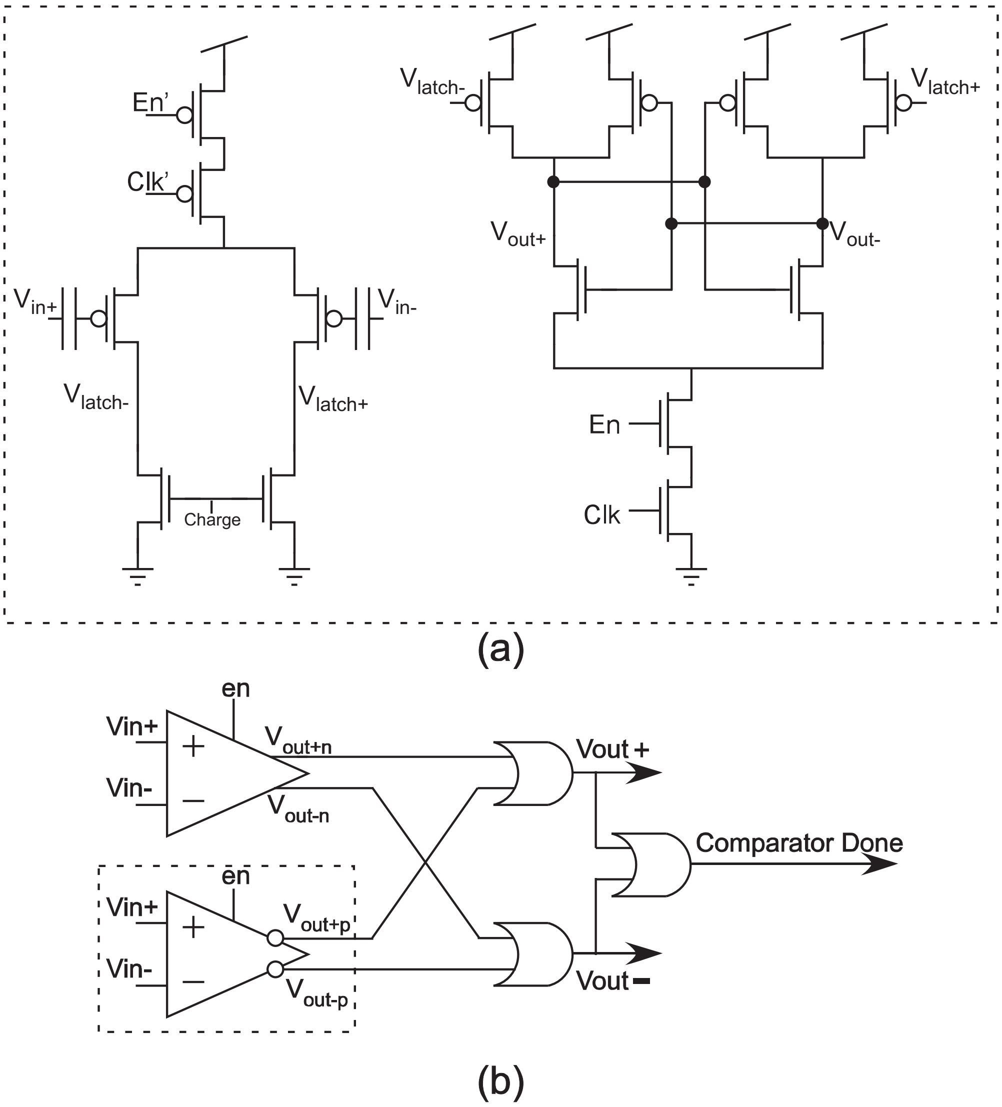


Figure 7.3: (a) A schematic diagram of the pFET-based portion of the comparator. The first half is a two-phase operated differential pair with an added enable gate. This half utilizes floating-gates for the differential pair, allowing an offset to be programmed to match the nFET-based portion of the comparator. The second half is a latch which utilizes the result of the differential pair to produce an output high on either the positive output terminal or the negative output terminal. (b) A top-level view of the comparator schematic. The boxed comparator is the pFET portion shown above. The unboxed portion is the nFET-based half.

input pairs.

Floating-gate devices operate similarly to regular CMOS devices, except they store a charge on their DC isolated node. This stored charge can be reprogrammed through Fowler-Nordheim tunneling or through hot-electron injection for a variety of purposes and applications. Here, we use it to program an input offset onto the pFET devices that will match that of the nFET devices. In doing so, we are able to subrange our comparators and still maintain a transfer curve of the ADC with no discontinuities.

7.4 Results

In this section, we will present results simulated in MATLAB. Although these are simulation based, they should be indicative of the power savings achievable through the use of an extrema SA-ADC across CMOS processes.

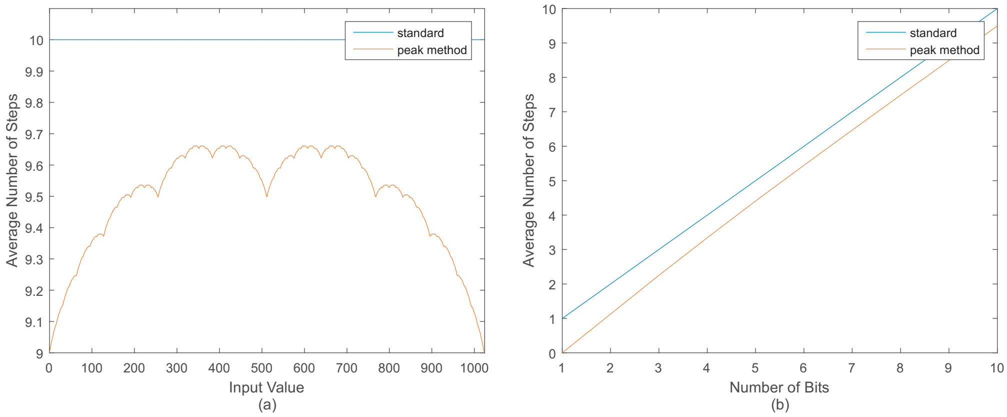


Figure 7.4: (a) The steps taken for any given input of a 10-bit system, averaged for every possible preceding value, in an SA-ADC as well as an extrema enhanced SA-ADC. As expected, this results in a constant 10 steps for the standard SA-ADC, since it does not depend upon previous input. (b) The average number of steps taken versus the resolution of the system in an SA-ADC as well as an extrema enhanced SA-ADC.

The first result presented here shows the number of operations we would expect the DAC and the comparator to perform for a linear distribution across all possible samples in a 10 bit system (Fig. 7.4(a)). We can see that, as expected, the normal SA-ADC must perform 10 operations. The SA-ADC saves an average of 0.6 steps. Observing the trend from 1 to 10 bit systems, we would expect this savings to become even smaller for higher-resolution systems (Fig. 7.4(b)).

This slight savings in steps is underwhelming, until one considers which steps are being skipped. The more significant the bit, the more likely it is to be the skipped step. Since SA-ADCs operate on binarily weighted capacitor arrays, we are often skipping the most power intensive steps. As seen in (Fig. 7.5(a)), this 0.6 step skip provides a %16.7 average

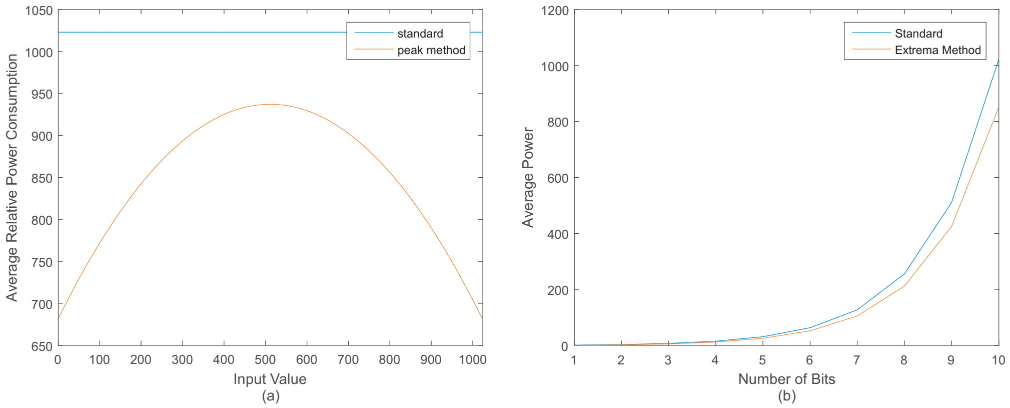


Figure 7.5: (a) The power consumption of a 10-bit system, averaged for every possible preceding value, in an SA-ADC as well as an extrema enhanced SA-ADC. This value is calculated by summing the total unit capacitance needed to generate words in the DAC. (b) The power savings versus the resolution of the system in an SA-ADC as well as an extrema-enhanced SA-ADC.

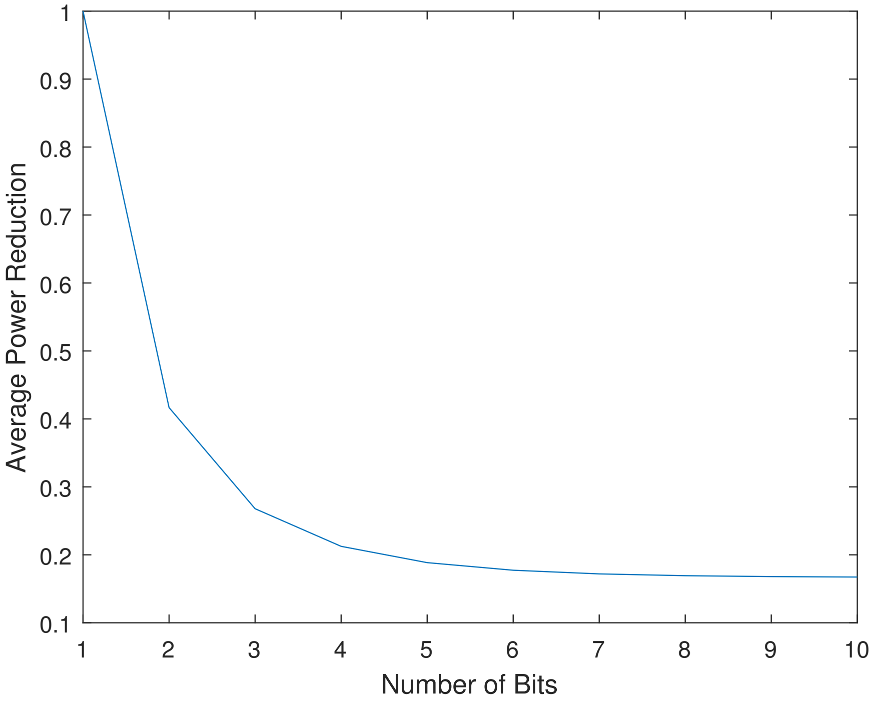


Figure 7.6: The percentage power saved by using the extrema-enhanced SA-ADC over a traditional SA-ADC versus the resolution of the system.

reduction in power for a 10 bit system. For higher resolution systems, we would expect this value to be only slight less (Fig. 7.5(b) and Fig. 7.6)

7.5 Conclusion

Presented here are improvements to a typical SA-ADC system. These include a comparator which takes advantage of the SAR algorithm so that it can span a full-scale range with minimal components. These also include a new algorithm for an extrema enhanced SA-ADC which allows power savings through the occasional skipping of steps.

Chapter 8

VCO-ADC with Floating-Gate Linearization

The voltage-controlled-oscillator analog-to-digital converter (VCO-ADC) is a highly digital quantizer, designed to take full advantage of compact digital design and power-efficient digital scaling. To first approximation, there is very little room for analog design within this particular topology. While this is largely true, this chapter will demonstrate how non-volatile analog memory can be used to address the largest issue within this topology – input-output linearity. The issue of linearity begins with the conversion of an input voltage to a current through a CMOS device – an inherently non-linear conversion. While many topological corrections to this exist, we will present here a method which is simpler, more compact, and more compatible with existing VCO-ADC designs.

8.1 Voltage-Controlled-Oscillators as ADC elements

Within the highly active area of analog to digital converters (ADC), voltage-controlled-oscillator (VCO) ADCs have shown numerous strengths. They are highly digital, which means not only does the power consumption scale well, but their resolution does as well. Their resolution can be scaled by extending the topology simply, in a manner similar to increasing the digits on a counter [93]. This digital scalability has made them valuable ADCs in their own right, but they are also used as the quantization stage in delta sigma ADCs

[94, 95]. Despite their numerous advantages, they suffer from poor linearity which limits their dynamic range, and in-turn their proliferation among resource constrained systems.

The inherent problem of poor linearization is often the focus of new VCO based ADC designs. Because the VCO is highly digital to begin with, digital correction techniques [96], including the use of look-up tables [97], are very popular. These digital methods are effective, but create processing requirements that may not be feasible in resource-constrained scenarios. Other methods have proposed improved linearity through the filtering operations associated with quantization, and have demonstrated success with both open loop [98] and closed loop designs [99]. However, while these filter improvements/changes improve the linearity of the device, they are incapable of extending the linear range completely from rail-to-rail.

Another class of changes include making changes to the topology of the VCO itself. Improvements in linearity through differential delay elements [100] or through the use of two separate VCOs to create a fully differential system [101] has been demonstrated. Subranging through the use of a successive approximation ADC in tandem with a VCO-ADC has also been explored [102]. Unfortunately, all of these designs add considerable energy and area overhead as well as complexity to the design.

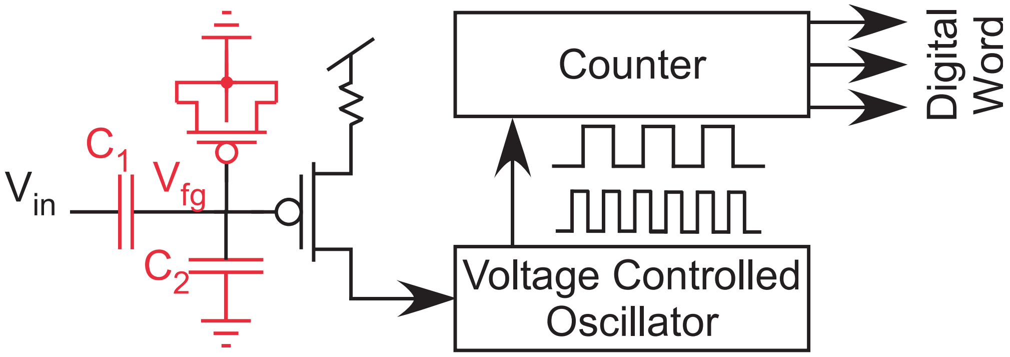


Figure 8.1: Block diagram of a VCO-ADC plus the proposed linearization method (shown in red). Linearization is achieved through capacitive division (the ratio of which is set by C_1/C_2) and the voltage set on V_{fg} through floating-gate charge programming techniques.

Instead of addressing the linearization issue by complicating the input quantity or through post-processing, our work takes a simpler, more passive route. The linearization issue occurs because the input voltage creates a current through a transistor, and it is this current which actually effects the frequency of the VCO. By remapping this input voltage to the region over which the current response of the transistor is linear (ie, by staying within the transistor's linear region), we will naturally have a linear response from the VCO and thus the ADC. We achieve this mapping through the use of a pre-programmed analog element, a floating-gate transistor, and through the use of a simple capacitive divider (Fig. 8.1).

The remainder of this chapter is organized as follows. Section II provides an overview of the system, including VCO-ADC operation as well as floating gate operation and programming. Section III provides details to the methods and limitations of our linearization technique. Finally, Section IV provides relevant results before concluding in Section V.

8.2 System Overview

The component-parts of this work include a standard VCO paired with a counter to form an ADC, as well as a floating-gate input stage (Fig. 8.1). Before demonstrating our linearization technique, we will first briefly review the operation of the VCO-ADC as well as the use of floating-gates as analog memory elements.

8.2.1 Principles of the VCO-ADC

Quantization within a VCO-ADC begins by taking an input voltage and converting it to a current. Typically, this is done by controlling a source-degenerated MOSFET channel current by its gate voltage. The issue of non-linearity begins here, where the current response of the MOSFET is linear only for a particular region of operation.

This channel current is then used to control an oscillator, which is most simply implemented by an odd number of current starved inverters. The odd number of stages in a feed-back configuration forces these digital elements to continually switch between logic-high and logic-low at a rate limited by the current starving. This oscillation can then be tracked by a counter. The output of this counter is then proportional to the input voltage and the time duration of operation. The time duration is controlled by resetting the counter with a regular clock. The result is a digital word output which is proportional only to the input voltage.

The period of the clock determines the sampling frequency. Sampling frequency in most

ADC architectures is often limited by the switching speed of individual transistors, and is thus inversely proportional to feature size. This limitation is also the case in the VCO-ADC, where the highest rate of oscillation determines the maximum sampling frequency. The oscillator has to be able to provide sufficient oscillations to force the counter into the all ones state before the sampling clock resets the counter:

$$f_s = \frac{1}{\frac{1}{f_{ROmax}} * 2^n} \tag{8.1}$$

where f_s is the frequency of the sampling clock, f_{ROmax} is the maximum frequency of the ring oscillator, and n is the number of bits of the counter (and by extension, the number of bits of the ADC).

8.2.2 Principles of Floating Gates

Floating-gate devices are most widely used in digital flash-memory applications, but they have also been shown to be extremely useful in the analog domain. One use of floating-gates is as non-volatile analog memory, in which they are programmed to store a particular charge, and thus provide a FET with a tunable threshold-voltage. By tuning the charge stored on the floating node, they have been used for offset removal [41] in differential pairs, threshold definition in flash ADCs [42], or for creating high-precision voltage references [103].

Modification of the stored charge is achieved through two means in this work: Fowler-Nordheim tunneling and hot electron injection. Fowler-Nordheim tunneling is the process by which charge is removed from the Floating Node. This process is the analog equivalent of erasure, allowing the transistor to react like its non-floating counterpart. Hot electron injection is the effect which we leverage to add charge to the floating node. These two methods together allow us to create a tunable transfer curve.

8.3 Linearization Technique

Linearization of the voltage-to-current-to-frequency response of the VCO-ADC is achieved through the use of a programmed floating gate input transistor and a capacitive divider. The

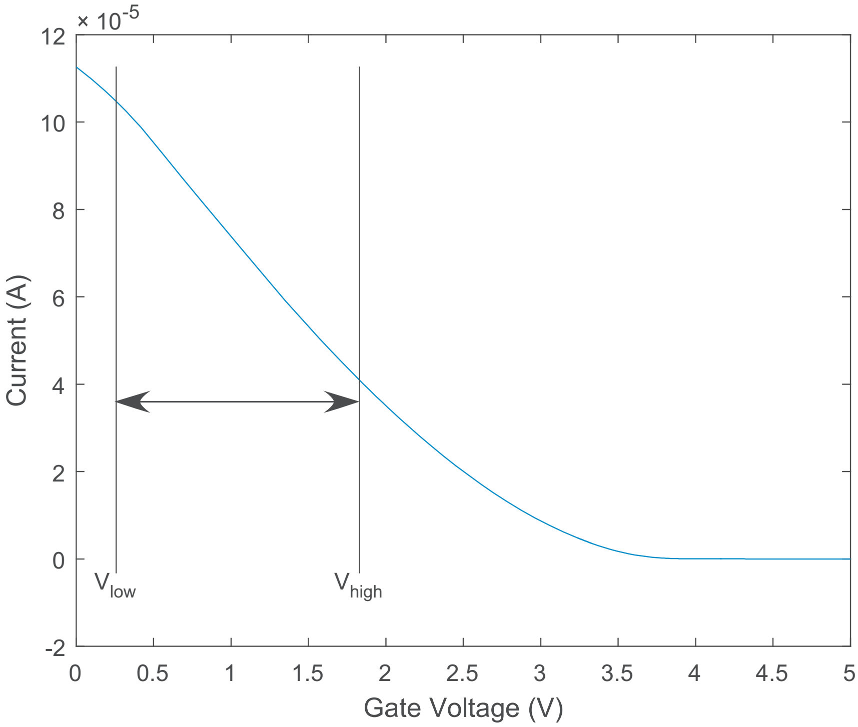


Figure 8.2: The inherent non-linearity of the gate-voltage to channel-current transformation. The linearization technique presented here allows a circuit designer to set a new mapping of input voltage, to the voltage seen by the transistor. This is done by setting a V_{low} through charge programming and a V_{high} through capacitive division.

design process begins by finding the voltages which create the boundaries of the linear region on the transistor's gate voltage to channel-current curve, V_{low} and V_{high} on Fig. 8.2.

V_{low} determines where the lowest input voltage will map to, as determined by the charge programmed on the floating-gate (Fig. 8.3). Since the floating-gate voltage is determined by

$$V_{fg} = \frac{Q}{C_T} + \frac{C_{cg}V_{in} + \sum C_x V_x}{C_T} \tag{8.2}$$

we can determine this charge by setting V_{in} equal to zero and solving for the programmed charge. This simplifies to programming the charge to be the desired V_{low}:

$$\frac{Q}{C_T} = V_{low} - \frac{V_{inLow}}{C_T} \tag{8.3}$$

where V_{low} is the voltage determined in Fig. 8.2, and V_{inLow} is the lowest input voltage that will be used.

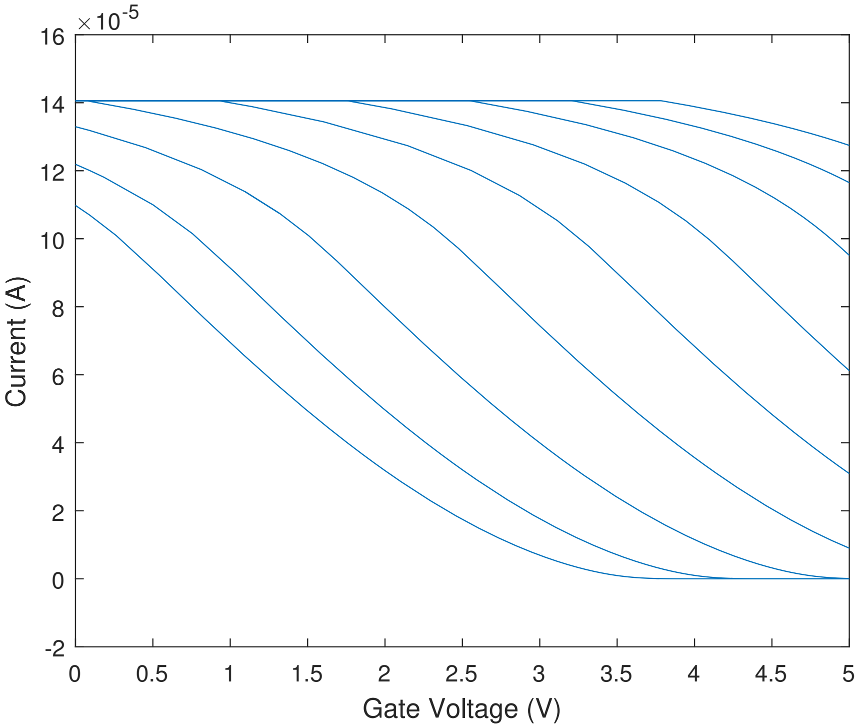


Figure 8.3: Increasing the initial charge on the floating-gate effectively shifts voltage-current curve to the left. This shifting allows us to set V_{low}.

V_{high} is the highest voltage which will be seen on the floating node, and is determined by capacitive division (Fig. 8.4). Simplifying the equation to account only for draw capacitance, Eq. 8.2 can be solved for this capacitive division

$$\frac{C1}{C2} = \frac{V_{high} - Q/C_T}{V_{inHigh}} \tag{8.4}$$

where V_{high} is the voltage determined in Fig. 8.2, and V_{inHigh} is the highest input voltage that will be used.

Finally, by first setting an initial charge and then setting a capacitive ratio, a highly linear voltage-to-current relationship can be attained, as seen in Fig. 8.5.

8.4 Results

The discussed system will be fabricated in a standard 0.35μm CMOS process (Fig. 8.6). Prior to fabrication, we can observe the simulation-based estimates of the performance in the context of other, state-of-the-art VCO-ADCs (Table 8.1). While the sampling frequency

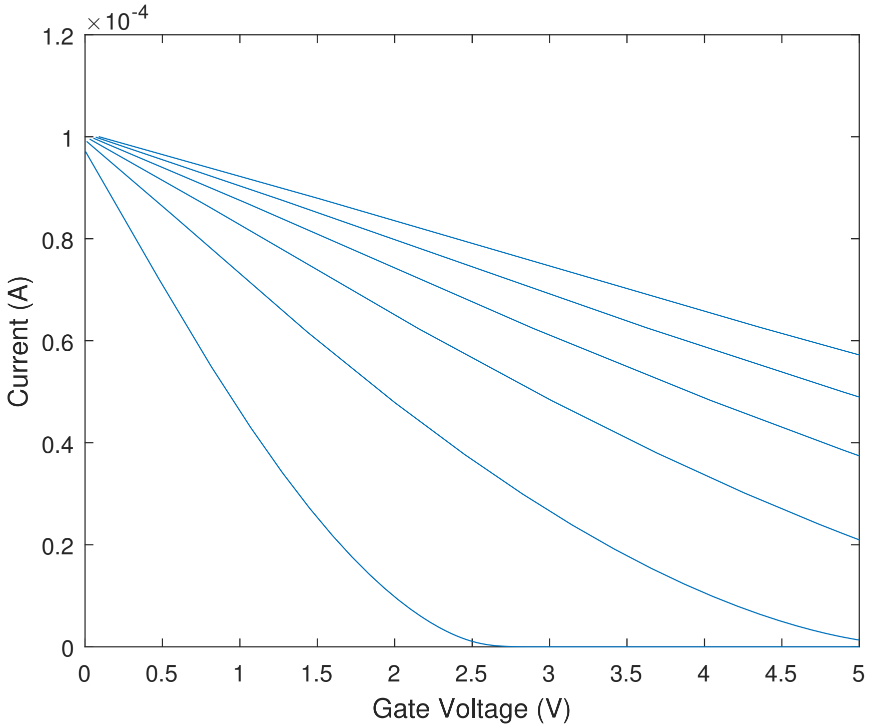


Figure 8.4: After choosing an initial charge, we can further linearize the curve through capacitive division. Higher capacitive division creates a smaller overall change in current. This reduced current swing allows us to set V_{high}.

of our system is considerably lower than the others, it is also striking how competitive our area consumption is despite being in a relatively older process.

Table 8.1: VCO-ADC Results Comparison

	This Work	**K. Reddy [104]**	**X. Xing [105]**	**A. Ghosh [106]**	**K. Ragab [107]**
Year	2017	2012	2015	2015	2017
Process (μm)	0.35	0.09	0.04	0.065	0.18
Power (mW)	0.137	16	2.57	8.2	4.8
Fs (MHz)	0.029	600	1600	1000	51.2
Area (mm^2)	0.18747	0.41	0.017	0.62	0.22

8.5 Conclusion

This work presented a simple and space-efficient method for linearizing the gate-voltage to channel-current relationship of a transistor which was used to linearize the input-output

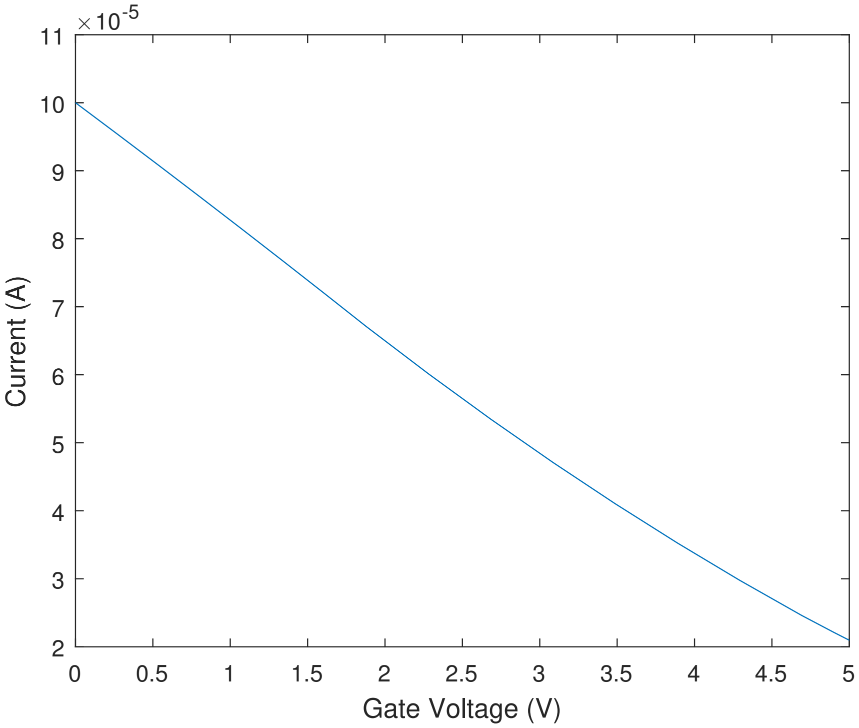


Figure 8.5: By setting V_{low} through charge programming and V_{high} through capacitive division, we can attain a highly-linear gate-voltage to channel-current transfer curve.

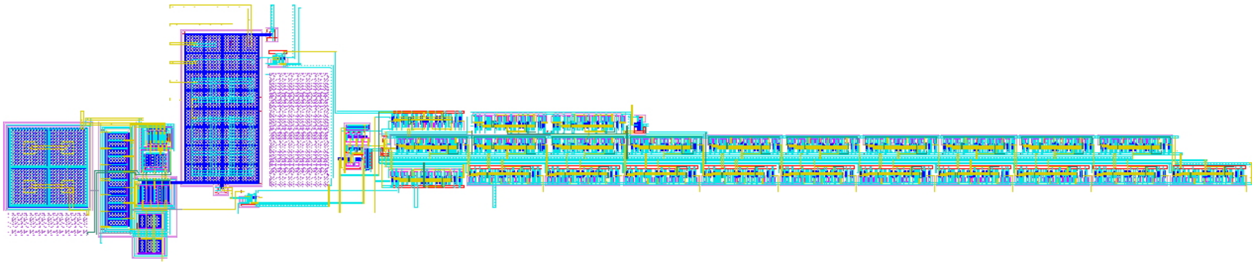


Figure 8.6: A new iteration of the VCO-ADC which will be fabricated in a 0.35μm CMOS process.

relationship of a VCO-ADC. Our simulated design was shown to be particularly competitive in area. The smaller area and simple linearization technique make this system suitable for a wide variety of systems or for use in combination with other linearization techniques.

Chapter 9

Summary and Future Work

The ubiquity of sensor-interfaced devices is climbing at an incredible rate – possibly because devices are most useful to us when they can interact with the outside world. This interaction, however, is one of the most costly operations within the domain of signal processing. Analog-to-digital conversion taxes the capabilities of sensor-interfaced devices. This work has studied techniques to mitigate the power consumption of these devices which fall outside the usual path of digital-scaling-driven power savings.

Digital-scaling has traditionally been relied upon for decreasing power expenditure, but it has diminishing returns in the analog and mixed-signal domains. Analog-to-digital converters (ADCs) are inherently mixed-signal devices, and it is a crux of this work that novel analog-driven techniques are necessary for continuing to realize power-savings within the field of ADC design. To that end, this work has presented techniques to adaptively sample and convert signals based upon their changing temporal characteristics – as analyzed by various analog-front ends.

Unfortunately, analog front-ends are plagued by problems common to all analog electronics. First, digital scaling leads to increases in transistor mismatch, causing a multitude of issues in analog design. Second, analog front-ends tend to be very application specific and require precise biasings. Floating-gate transistors (also known as analog memory or flash memory) provide an answer to many of these problems by allowing the circuit designer to 'program' transistor-level adjustments post-fabrication. This work has provided a step toward the proliferation of these floating-gate device by demonstrating a working simulation model and by demonstrating its efficacy in fabricated circuitry. This simulation has been verified for $0.5\mu m$ and $0.35\mu m$ CMOS processes, but newer processes will likely pose some interesting challenges. The extremely thin oxides of sub-100nm processes may yield unique charge retention and programming characteristics for floating-gate devices. However, the basic topology of the presented macro-model should be a sufficient foundation for developing accurate simulations of these newer processes.

In addition to the floating-gate model, the other mathematically based contribution of this work is in the area of asynchronous sampling – namely extrema sampling. Extrema sampling has been demonstrated both in simulation as well as in silicon in this work, and has shown the ability to conserve system resources by adapting to a target signal.

Detecting and capturing these extrema values in silicon is a difficult task, particularly at higher frequencies. This difficulty is owed to the fact that extrema detection is essentially a matter of finding the zero crossing of the first derivative. Derivative operations are known to be a very frequency limited operation [72]. Development of an extrema-detector circuit with a wider range of operating frequencies would provide a large step to increasing the proliferation of extrema-based devices, and would be an excellent area for future research.

Reconstruction of these asynchronously-sampled signals has been touched on, but is another prime candidate for future research. It is likely that some of the most efficient and interesting reconstruction methods will be developed for individual applications. That is to say, taking advantage of understanding the form of the sampled signal ahead of time should provide a great benefit in designing its reconstruction.

The combination of these foundational circuit building blocks enabled the construction of various systems including analog front-ends, temperature compensation circuits, and data converters. The common threads between these systems were low-power, floating-gate enabled, and event-triggered operation.

While these works have made advances in asynchronous data conversion, the area remains open to many more investigations. A natural extension of this work would be to place it in applications which require more advanced analog signal classification – possibly through the simultaneous analysis of signals from different types of sensors. For example, the sleep apnea database [87] includes ECG signals, blood pressure signals, and many other signals along with annotations which denote the occurrence of a respiratory episode. An interesting project would be to simultaneously monitor these varied signals in a reconfigurable analog front-end, predict the onset of a respiratory episode, and trigger data conversion based on that prediction.

Beyond the previously mentioned areas of possible future research, the use of self-adapting and learning circuits within analog front-ends and data conversion systems is a particularly untapped area of research. As more of these learning circuits are realized in forms that fit energy-constrained systems, they may yield results as fruitful as the first analog wake-up detectors.

References

[1] 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.

[2] 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 monioring of exceptional events," in *ACM Sensys*, 2007.

[3] T. Feng, K. Aono, T. Covassin, and S. Chakrabartty, "Self-powered monitoring of repeated head impacts using time-dilation energy measurement circuit," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 9, no. 2, pp. 217–226, April 2015.

[4] X. Zhang and Y. Lian, "A 300-mv 220-nw event-driven adc with real-time qrs detection for wearable ecg sensors," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 8, no. 6, pp. 834–843, Dec 2014.

[5] H. M. Wang, Y. L. Lai, M. C. Hou, S. H. Lin, B. S. Yen, Y. C. Huang, L. C. Chou, S. Y. Hsu, S. C. Huang, and M. Y. Jan, "A ±6ms-accuracy, 0.68mm^2 and 2.21μw qrs detection asic," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, May 2010, pp. 1372–1375.

[6] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, Jan 1949.

[7] G. Fant, *Acoustic Theory of Speech Production.* Berlin:De Gruyter, 1971.

[8] M. Unser, "Sampling-50 years after shannon," *Proceedings of the IEEE*, vol. 88, no. 4, pp. 569–587, April 2000.

[9] ——, "Splines: a perfect fit for signal and image processing," *IEEE Signal Processing Magazine*, vol. 16, no. 6, pp. 22–38, Nov 1999.

[10] M. Stephane, *A Wavelet Tour of Signal Processing (Third Edition).* Academic Press, 2009.

[11] B. Murmann. ADC performance survey 1997-2016. [Online]. Available: http://web.stanford.edu/ murmann/adcsurvey.html.

[12] A. Varzaghani, A. Kasapi, D. Loizos, S.-H. Paik, S. Verma, S. Zogopoulos, and S. Sidiropoulos, "A 10.3-gs/s, 6-bit flash ADC for 10g ethernet applications," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 12, pp. 3038–3048, Dec 2013.

[13] M. Miyahara, I. Mano, M. Nakayama, K. Okada, and A. Matsuzawa, "A 2.2gs/s 7b 27.4mw time-based folding-flash ADC with resistively averaged voltage-to-time amplifiers," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, Feb 2014, pp. 388–389.

[14] Y. Miyahara, M. Sano, K. Koyama, T. Suzuki, K. Hamashita, and B.-S. Song, "Adaptive cancellation of gain and nonlinearity errors in pipelined ADCs," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, Feb 2013, pp. 282–283.

[15] S. Danesh, J. Hurwitz, K. Findlater, D. Renshaw, and R. Henderson, "A reconfigurable 1 gsps to 250 msps, 7-bit to 9-bit highly time-interleaved counter ADC with low power comparator design," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 3, pp. 733–748, March 2013.

[16] H.-Y. Tai, Y.-S. Hu, H.-W. Chen, and H.-S. Chen, "A 0.85fj/conversion-step 10b 200ks/s subranging SAR ADC in 40nm CMOS," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, Feb 2014, pp. 196–197.

[17] S. Lee, A. Chandrakasan, and H.-S. Lee, "A 1gs/s 10b 18.9mw time-interleaved SAR ADC with background timing-skew calibration," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, Feb 2014, pp. 384–385.

[18] P. Harpe, Y. Zhang, G. Dolmans, K. Philips, and H. de Groot, "A 7-to-10b 0-to-4ms/s flexible SAR ADC with 6.5-to-16fj/conversion-step," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, Feb 2012, pp. 472–474.

[19] P. Harpe, E. Cantatore, and A. van Roermund, "A 2.2/2.7fj/conversion-step 10/12b 40ks/s SAR ADC with data-driven noise reduction," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, Feb 2013, pp. 270–271.

[20] B. E. Jonsson, "An empirical approach to finding energy efficient ADC architectures," in *IEEE International Workshop on ADC Modelling, Testing and Data Converter Analysis and Design*, 2011.

[21] B. Murmann, "A/d converter trends: Power dissipation, scaling and digitally assisted architectures," in *2008 IEEE Custom Integrated Circuits Conference*, Sept 2008, pp. 105–112.

[22] R. Sarpeshkar, "Analog versus digital: Extrapolating from electronics to neurobiology," *Neuroal Computation*, vol. 10, pp. 1601–1608, 1998.

[23] 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.

[24] B. Kelly, B. Rumberg, and D. Graham, "An ultra-low-power analog memory system with an adaptive sampling rate," in *Proc. of IEEE MWSCAS*, Aug. 2012, pp. 302–305.

[25] B. M. Kelly, B. D. Rumberg, and D. W. Graham, "Compressed sampling and memory," Jun. 14 2016, uS Patent 9,367,079.

[26] B. Rumberg, D. Graham, S. Clites, B. Kelly, M. Navidi, A. Dilello, and V. Kulathumani, "RAMP: Accelerating wireless sensor hardware design with a reconfigurable analog/mixed-signal platform," in *Proc. of ACM/IEEE Inform. Process. in Sensor Networks*, 2015, pp. 47–58.

[27] H. Yoo and C. van Hoof, *Bio-Medical CMOS ICs.* Springer, 2010.

[28] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.

[29] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, Dec 2006.

[30] A. Wang, F. Lin, Z. Jin, and W. Xu, "Ultra-low power dynamic knob in adaptive compressed sensing towards biosignal dynamics," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, no. 3, pp. 579–592, June 2016.

[31] J. Mark and T. Todd, "A nonuniform sampling approach to data compression," *IEEE Transactions on Communications*, vol. 29, no. 1, pp. 24–32, Jan 1981.

[32] M. Greitans, R. Shavelis, L. Fesquet, and T. Beyrouthy, "Combined peak and level-crossing sampling scheme," in *Proc. of the International Conference on Sampling Theory and Applications*, May 2011.

[33] 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.

[34] 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.

[35] E. Mackensen and C. Muller, "Implementation of reconfigurable micro-sensor interfaces utilizing FPAAs," in *IEEE Sensors*, Nov. 2005, pp. 1064–1067.

[36] S. George, S. Kim, S. Shah, J. Hasler, M. Collins, F. Adil, R. Wunderlich, S. Nease, and S. Ramakrishnan, "A programmable and configurable mixed-mode fpaa soc," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 6, pp. 2253–2261, June 2016.

[37] B. M. Kelly, B. Rumberg, D. W. Graham, and V. Kulathumani, "Reconfigurable analog signal processing for wireless sensor networks," in *Circuits and Systems (MWSCAS), 2013 IEEE 56th International Midwest Symposium on.* IEEE, 2013, pp. 221–224.

[38] B. M. Kelly, B. Rumberg, D. W. Graham, V. Kulathumani, S. Clites, A. Dilello, and M. M. Navidi, "Ramp: accelerating wireless sensor hardware design with a reconfigurable analog/mixed-signal platform," in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks.* ACM, 2015, pp. 402–403.

[39] B. Rumberg, B. M. Kelly, D. W. Graham, and V. Kulathumani, "Demo abstract: netamorph: field-programmable analog arrays for energy-efficient sensor networks," in *Proceedings of the 12th international conference on Information processing in sensor networks.* ACM, 2013, pp. 309–310.

[40] P. Lajevardi, A. P. Chandrakasan, and H. S. Lee, "Zero-crossing detector based reconfigurable analog system," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 11, pp. 2478–2487, Nov 2011.

[41] F. Adil, G. Serrano, and P. Hasler, "Offset removal using floating-gate circuits for mixed-signal systems," in *Proc. of Southwest Symposium on Mixed-Signal Design*, Feb. 2003, pp. 190–195.

[42] Y. L. Wong, M. H. Cohen, and P. A. Abshire, "A 750-mhz 6-b adaptive floating-gate quantizer in 0.35μm CMOS," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 7, pp. 1301–1312, July 2009.

[43] D. W. Graham, E. Farquhar, B. Degnan, C. Gordon, and P. Hasler, "Indirect programming of floating-gate transistors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 5, pp. 951–963, May 2007.

[44] S.-C. Liu, T. Delbruck, G. Indiveri, A. Whatley, and R. Douglas, *Event-based neuromorphic systems.* Wiley, February 2016.

[45] M. Lenzlinger and E. H. Snow, "Fowler-nordheim tunneling into thermally grown sio2," *IEEE Transactions on Electron Devices*, vol. 15, no. 9, pp. 686–686, Sep 1968.

[46] K. Hasnat, C. F. Yeap, S. Jallepalli, W. K. Shih, S. A. Hareland, V. M. Agostinelli, A. F. Tasch, and C. M. Maziar, "A pseudo-lucky electron model for simulation of electron gate current in submicron nmosfet's," *IEEE Transactions on Electron Devices*, vol. 43, no. 8, pp. 1264–1273, Aug 1996.

[47] D. Rzepka and M. Miskowicz, "Recovery of varying-bandwidth signal from samples of its extrema," in *Signal Processing: Algorithms, Architectures, Arrangements, and Applications*, Sept 2013, pp. 143–148.

[48] L. Kull, T. Toifl, M. Schmatz, P. Francese, C. Menolfi, M. Braendli, M. Kossel, T. Morf, T. Andersen, and Y. Leblebici, "A 3.1mw 8b 1.2gs/s single-channel asynchronous SAR ADC with alternate comparators for enhanced speed in 32nm digital soi CMOS," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, Feb 2013, pp. 468–469.

[49] P. Harpe, C. Zhou, X. Wang, G. Dolmans, and H. de Groot, “A 30fj/conversion-step 8b 0-to-10ms/s asynchronous SAR ADC in 90nm CMOS,” in *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, Feb 2010, pp. 388–389.

[50] M. Trakimas and S. Sonkusale, “An adaptive resolution asynchronous ADC architecture for data compression in energy constrained sensing applications,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 5, pp. 921–934, May 2011.

[51] B. Swann, B. Blalock, L. Clonts, D. Binkley, J. Rochelle, E. Breeding, and K. Baldwin, “A 100-ps time-resolution CMOS time-to-digital converter for positron emission tomography imaging applications,” *IEEE Journal of Solid-State Circuits*, vol. 39, no. 11, pp. 1839–1852, Nov 2004.

[52] V.-C. Chen and L. Pileggi, “A 69.5mw 20gs/s 6b time-interleaved ADC with embedded time-to-digital calibration in 32nm CMOS SOI,” in *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, Feb 2014, pp. 380–381.

[53] K. Rahimi, C. Diorio, C. Hernandez, and M. D. Brockhausen, “A simulation model for floating-gate mos synapse transistors,” in *Proc. of IEEE ISCAS*, vol. 2, 2002, pp. II–532–II–535 vol.2.

[54] J. Gray, R. Robucci, and P. Hasler, “The design and simulation model of an analog floating-gate computational element for use in large-scale analog reconfigurable systems,” in *2008 51st Midwest Symposium on Circuits and Systems*, Aug 2008, pp. 253–256.

[55] P. Hasler, A. Basu, and S. Kozil, “Above threshold pfet injectionmodeling intended for programmingfloating-gate systems,” in *2007 IEEE International Symposium on Circuits and Systems*, May 2007, pp. 1557–1560.

[56] A. F. Mondragon-Torres, M. C. Schneider, and E. Sanchez-Sinencio, “Extraction of electrical parameters of floating gate devices for circuit analysis, simulation, and design,” in *Proc. of IEEE MWSCAS*, vol. 1, Aug 2002, pp. I–311–14 vol.1.

[57] L. Larcher, P. Pavan, S. Pietri, L. Albani, and A. Marmiroli, “A new compact dc model of floating gate memory cells without capacitive coupling coefficients,” *IEEE Transactions on Electron Devices*, vol. 49, no. 2, pp. 301–307, Feb 2002.

[58] S. J. Rapp, K. R. McMillan, and D. W. Graham, “Spice-compatible modelling technique for simulating floating-gate transistors,” *Electronics Letters*, vol. 47, no. 8, pp. 483–485, April 2011.

[59] E. Rodriguez-Villegas, M. Jimenez, and R. G. Carvajal, “On dealing with the charge trapped in floating- gate mos (fgmos) transistors,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54, no. 2, pp. 156–160, Feb 2007.

[60] A. Veeravalli, E. Sanchez-Sinencio, and J. Silva-Martinez, "A CMOS transconductance amplifier architecture with wide tuning range for very low frequency applications," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 6, pp. 776–781, Jun 2002.

[61] L. Yin, S. H. K. Embabi, and E. Sanchez-Sinencio, "A floating-gate mosfet d/a converter," in *Proc. of IEEE ISCAS*, vol. 1, Jun 1997, pp. 409–412.

[62] B. Rumberg and D. W. Graham, "Efficiency and reliability of fowler-nordheim tunnelling in CMOS floating-gate transistors," *Electronics Letters*, vol. 49, no. 23, pp. 1484–1486, Nov 2013.

[63] P. Hasler, B. A. Minch, and C. Diorio, "Adaptive circuits using pfet floating-gate devices," in *Proc. of IEEE VLSI*, Mar 1999, pp. 215–229.

[64] C. Diorio, "A p-channel mos synapse transistor with self-convergent memory writes," in *IEEE transactions on Electron Devices*, Feb 2000, pp. 464–472.

[65] C. Huang, P. Sarkar, and S. Chakrabartty, "Rail-to-rail, linear hot-electron injection programming of floating-gate voltage bias generators at 13-bit resolution," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 11, pp. 2685–2692, Nov 2011.

[66] A. Bandyopadhyay, G. J. Serrano, and P. Hasler, "Adaptive algorithm using hot-electron injection for programming analog computational memory elements within 0.2% of accuracy over 3.5 decades," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 9, pp. 2107–2114, Sept 2006.

[67] B. Rumberg and D. W. Graham, "A floating-gate memory cell for continuous-time programming," in *Proc. of IEEE MWSCAS*, Aug 2012, pp. 214–217.

[68] E. Sackinger and W. Guggenbuhl, "An analog trimming circuit based on a floating-gate device," *IEEE Journal of Solid-State Circuits*, vol. 23, no. 6, pp. 1437–1440, Dec 1988.

[69] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. Wright, "Convergence properties of the nelder-mead simplex method in low dimensions," *SIAM Journal of Optimization*, vol. 9, no. 1, pp. 112–147, 1998.

[70] K. Rahimi, C. Diorio, C. Hernandez, and M. D. Brockhausen, "A simulation model for floating-gate mos synapse transistors," in *IEEE International Symposium on Circuits and Systems*, vol. 2, 2002, pp. 532–535.

[71] B. M. Kelly and D. W. Graham, "An asynchronous ADC with reconfigurable analog pre-processing," in *Proc. of IEEE ISCAS*, May 2016, pp. 1062–1065.

[72] C. A. Mead, *Analog VLSI and Neural Systems.* Addison-Wesley, 1989.

[73] Y. Lyubarskii and W. Madych, "The recovery of irregularly sampled band limited functions via tempered splines," *Journal of Functional Analysis*, vol. 125, no. 1, pp. 201 – 222, 1994.

[74] D. Rzepka and M. Miskowicz, "Recovery of varying-bandwidth signal from samples of its extrema," in *Signal Processing: Algorithms, Architectures, Arrangements, and Applications*, Sept. 2013, pp. 143–148.

[75] P. Martnez-Nuevo, S. Patil, and Y. Tsividis, "Derivative level-crossing sampling," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 1, pp. 11–15, Jan 2015.

[76] B. Rumberg and D. W. 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.

[77] S. Ravindran, P. Smith, D. Graham, V. Duangudom, D. Anderson, and P. Hasler, "Towards low-power on-chip auditory processing," *EURASIP J. Applied Sig. Process.*, vol. 2005, no. 7, pp. 1082–1092, May 2005.

[78] 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 J. Solid-State Circuits*, vol. 45, no. 9, pp. 1781–1794, Sept. 2010.

[79] V. Srinivasan, G. Serrano, C. M. Twigg, and P. Hasler, "A floating-gate-based programmable CMOS reference," *IEEE Trans. Circuits Syst. I*, vol. 55, no. 11, pp. 3448–3456, Dec. 2008.

[80] M. Gu and S. Chakrabartty, "Subthreshold, varactor-driven CMOS floating-gate current memory array with less than 150-ppm/° k temperature sensitivity," *IEEE J. Solid-State Circuits*, vol. 47, no. 11, pp. 2846–2856, Nov. 2012.

[81] L. Zhou and S. Chakrabartty, "A continuous-time varactor-based temperature compensation circuit for floating-gate multipliers and inner-product circuits," in *IEEE Int. Symp. on Circuits and Syst.*, May 2015.

[82] B. Rumberg and D. Graham, "Sub-microwatt analog VLSI trainable pattern classifier," *Proc. IEEE MWSCAS*, pp. 214–217, Aug 2012.

[83] K. Kozmin, J. Johansson, and J. Delsing, "Level-crossing ADC performance evaluation toward ultrasound application," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 8, pp. 1708–1719, Aug 2009.

[84] D. Schinkel, E. Mensink, E. Klumperink, E. van Tuijl, and B. Nauta, "A double-tail latch-type voltage sense amplifier with 18ps setup+hold time," in *Proc. of IEEE ISSCC*, Feb. 2007, pp. 314–605.

[85] G. Roberts and M. Ali-Bakhshian, "A brief introduction to time-to-digital and digital-to-time converters," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 3, pp. 153–157, March 2010.

[86] C. J. DeLuca, "The use of surface electromyography in biomechanics," *Journal of Applied Biomechanics*, vol. 13, pp. 135–163, 1997.

[87] A. L. Goldberger *et al.*, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[88] C. I. Ieong, P. I. Mak, C. P. Lam, C. Dong, M. I. Vai, P. U. Mak, S. H. Pun, F. Wan, and R. P. Martins, "A 0.83-μw qrs detection processor using quadratic spline wavelet transform for wireless ecg acquisition in 0.35-μm cmos," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 6, no. 6, pp. 586–595, Dec 2012.

[89] R. A. Abdallah and N. R. Shanbhag, "A 14.5 fj/cycle/k-gate, 0.33 v ecg processor in 45nm cmos using statistical error compensation," in *Proceedings of the IEEE 2012 Custom Integrated Circuits Conference*, Sept 2012, pp. 1–4.

[90] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita, "A line in the sand: a wireless sensor network for target detection, classification, and tracking," *Computer Networks*, vol. 46, no. 5, pp. 605 – 634, 2004, military Communications Systems and Technologies.

[91] K. Chen, Z. Yang, L. Hoang, J. Weiland, M. Humayun, and W. Liu, "An integrated 256-channel epiretinal prosthesis," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 9, pp. 1946–1956, Sept 2010.

[92] 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.

[93] W. Jiang, V. Hokhikyan, H. Chandrakumar, V. Karkare, and D. Markovic, "A 50mv linear-input-range VCO-based neural-recording front-end with digital nonlinearity correction," in *Proc. of IEEE ISSCC*, Jan 2016, pp. 484–485.

[94] G. Taylor and I. Galton, "A mostly-digital variable-rate continuous-time delta-sigma modulator ADC," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 12, pp. 2634–2646, Dec 2010.

[95] W. Yu, J. Kim, K. Kim, and S. Cho, "A time-domain high-order mash $\Delta\Sigma$ ADC using voltage-controlled gated-ring oscillator," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 4, pp. 856–866, April 2013.

[96] K. Ragab and N. Sun, "A 12b enob, 2.5mhz-bw, 4.8mw VCO-based 0-1 MASH ADC with direct digital background nonlinearity calibration," in *Proc. of IEEE CICC*, Sept 2015, pp. 1–4.

[97] J. McNeill, R. Majidi, J. Gong, and C. Liu, "Lookup-table-based background linearization for VCO-based ADCs," in *Proceedings of the IEEE 2014 Custom Integrated Circuits Conference*, Sept 2014, pp. 1–4.

[98] P. K. Sharma and M. S. W. Chen, "A 6b 800ms/s 3.62mw nyquist ac-coupled VCO-based ADC in 65nm CMOS," in *Proc. of IEEE CICC*, Sept 2013, pp. 1–4.

[99] P. Prabha, S. J. Kim, K. Reddy, S. Rao, N. Griesert, A. Rao, G. Winter, and P. K. Hanumolu, "A VCO-based current-to-digital converter for sensor applications," in *Proceedings of the IEEE 2014 Custom Integrated Circuits Conference*, Sept 2014, pp. 1–4.

[100] A. Babaie-Fishani and P. Rombouts, "Highly linear VCO for use in VCO-ADCs," *Electronics Letters*, vol. 52, no. 4, pp. 268–270, 2016.

[101] T. He, Y. Du, Y. Jiang, S. W. Sin, S. P. U, and R. P. Martins, "A dual-VCO-based quantizer with highly improved linearity and enlarged dynamic range," in *Proc. of IEEE MWSCAS*, Aug 2011, pp. 1–4.

[102] A. Sanyal, K. Ragab, L. Chen, T. R. Viswanathan, S. Yan, and N. Sun, "A hybrid SAR-VCO Δ Σ ADC with first-order noise shaping," in *Proceedings of the IEEE 2014 Custom Integrated Circuits Conference*, Sept 2014, pp. 1–4.

[103] B. K. Ahuja, H. Vu, C. A. Laber, and W. H. Owen, "A very high precision 500-na CMOS floating-gate analog voltage reference," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 12, pp. 2364–2372, Dec 2005.

[104] K. Reddy, S. Rao, R. Inti, B. Young, A. Elshazly, M. Talegaonkar, and P. K. Hanumolu, "A 16-mw 78-db sndr 10-mhz bw ct *deltasigma* adc using residue-cancelling vco-based quantizer," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 12, pp. 2916–2927, Dec 2012.

[105] X. Xing and G. G. E. Gielen, "A 42 fj/step-fom two-step vco-based delta-sigma adc in 40 nm cmos," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 3, pp. 714–723, March 2015.

[106] A. Ghosh and S. Pamarti, "Linearization through dithering: A 50 mhz bandwidth, 10-b enob, 8.2 mw vco-based adc," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 9, pp. 2012–2024, Sept 2015.

[107] K. Ragab and N. Sun, "A 12-b enob 2.5-mhz bw vco-based 0-1 mash adc with direct digital background calibration," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 2, pp. 433–447, Feb 2017.