2019

Low-Power Reconfigurable Sensing Circuitry for the Internet-of-Things Paradigm

Alexander DiLello
West Virginia University, adilello@mix.wvu.edu

Low-Power Reconfigurable Sensing Circuitry for the Internet-of-Things Paradigm

Alexander Todd DiLello

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

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2019

Abstract

Low-Power Reconfigurable Sensing Circuitry for the Internet-of-Things Paradigm

Alexander Todd DiLello

With ubiquitous wireless communication via Wi-Fi and nascent 5th Generation mobile communications, more devices – both smart and traditionally "dumb" – will be interconnected than ever before. This burgeoning trend is referred to as the Internet-of-Things. These new sensing opportunities place a larger burden on the underlying circuitry that must operate on finite battery power and/or within energy-constrained environments. New developments of low-power reconfigurable analog sensing platforms like field-programmable analog arrays (FPAAs) present an attractive sensing solution by processing data in the analog domain while staying flexible in design. This work addresses some of the contemporary challenges of low-power wireless sensing via traditional application-specific sensing and with FPAAs. A large emphasis is placed on furthering the development of FPAAs by making them more accessible to designers without a strong integrated-circuit background – much like FPGAs have done for digital designers.

Acknowledgments

This material is based upon the generous support by the National Science Foundation under Award No. CNS-1148815, Department of Energy's National Energy Technology Laboratory under Federal Grant DE-FE0012383, and West Virginia University's Provost Fellowship.

Contents

List of Figures

Chapter 1

Introduction

Wireless communication is becoming more ubiquitous with the deployment of Wi-Fi and nascent 5th Generation mobile communications. Furthermore, the bandwidth capacity has never been larger. This bounty of wireless communication has created an opportunity to connect together "dumb" devices – that is devices that were not traditionally networked and communicating with other devices – and make them "smart." This burgeoning trend is referred to as the Internet of Things (IoT) and is most visibly manifested to the common consumer in smart-home technology (e.g. thermostats, speakers, refrigerators) and smart-car technology.

In connecting many devices, there is an exchange of information stemming from the devices' sensors or their remote control of actuators. In some application spaces like the smart-home realm, there is the accessibility of a constant power source to accomplish this type of computing. But for other applications areas that exist in energy-constrained environments like agricultural sensing, the power consumption of the device is a large component in what determines its viability. Energy-constrained environments rely on a battery of finite energy. For these remote sensing applications, there is a need for energy-conscious designs to complement the developments made in wireless communication.

The objective of this work is to addresses the sensing circuitry challenges for the Internet of Things paradigm. It will advocate for a general-purpose, low-power, reconfigurable analog circuitry solution. Beyond the immediate benefits for the aforementioned sensing applications, developments in this field could have large impacts on integrated analog design because of the possibility of integrated analog designs done post-fabrication. Longer-term implications for this work would be a contribution to the democratization of analog integrated circuit design akin to what FPGAs have done to integrated digital design. Further background and context is expanded upon below.

The conventional development of low-power sensing circuity has come from two different sources — a digital solution and an analog solution. The former has reduced power consumption by downscaling its feature sizes which have allowed its voltage supply rail to shrink down to 0.75V in a nascent 7nm process [3]. Furthermore, the power savings have been compounded with supply voltage reduction because it's characterized by a quadratic relationship in $P = CV^2f$. However, downscaling has not been following Moore's law within the past five years and it is showing diminishing returns due to semiconductor processing complications at sub 100nm feature sizes. For example, from 2006-2011, Intel had been following a schedule for downscaling the process about every two years (i.e. "tick-tock" development cycle) from 65nm to 22nm for their popular desktop CPUs. However, from 2011 to early 2019 (at the time of this writing), there has only been one downscale from 22nm to 14nm process. While there are optimizations in design to save power, the supply rail is not likely to drop as precipitously as it did from 5V to 0.75V in the next decade.

Low-power sensing in the analog domain has also benefited from downscaling, but the most dominant component of power-reduction has come from keeping computation in its original domain to save power. It's natural to try to keep most of the sensing in the analog realm as long as possible because the world that is being sensed is inherently analog. At some point in the communication chain, the analog values need to be converted to their digital equivalency for a microprocessor. However, being the conversion process is power-hungry, there are opportunities for power savings by employing analog front-ends to give discretion on what data to convert. An example application for this analog-digital interface to be applied is speech sensing. One power-savings opportunity would be to convert only speech data but not silence and non-speech audio. This could be accomplished with an analog front-end speech detector that listens for speech events while the digital circuitry stays in a low-power sleep mode when speech is not occurring. When speech is detected, the digital portions can then perform the digital conversion and transmission.

Like the speech detection example, the most power savings can be achieved by minimizing the use of a general-purpose microprocessor and offloading a portion of the microprocessor computation to the analog domain. The traditional method of accomplishing this is choosing an application-specific integrated circuit (ASIC) once the application space has been determined. ASICs focus on one specific task and do it very efficiently. However, if at any point during the design phase that the priorities change or the sensing fundamentally changes (e.g. changing a resistive-type temperature sensor switching a voltage-based sensor), the ASIC may not be able to accommodate the change. For battery-powered sensing in the IoT realm, there is a need for efficient sensing like that of ASICs, but there lacks a solution in design flexibility like that of field-programmable gate arrays (FPGAs) in the digital realm.

FPGAs provide application-specific integrated design, as well as flexibility in its reporgrammability. In the past, there have not been many platforms that have this ability to reconfigure integrated analog circuity. However, recent developments of field-programmable analog arrays (FPAAs) has brought this closer to a reality [4, 5, 6, 7]. FPAAs provide reconfigurability of integrated analog components (active and passive) at an ultra-low power consumption (typical applications in low 10's of μW) [8] – a great option to be used as analog front-ends for IoT wireless sensing applications. Furthermore, if the wireless aspect of these potential sensing devices is leveraged, the FPAAs can be reconfigured in the field.

FPAAs fulfill this need for ultra low power sensing for wireless devices. Its reconfigurability and low-power operation can be attributed to large sets of floating-gate FG transistors. For analog applications, FGs are used as programmable current sources for a variety of different circuit applications. With all the benefits, FGs do not come without their challenges. Currently, the low-power operation of FGs has an exponential relationship with temperature change, which becomes more of a challenge on an FG-dense FPAA platform. Similar issues arise in creating repeatable and accurate programming of FGs within an array of FGs. For both temperature and programming, reducing overhead and obviating external components equipment becomes a goal for an FPAA platform to succeed as an in-the-field circuit solution for wireless sensing. In the rest of this work, I set out to address these issues and contribute to improving wireless sensing for energy-constrained applications.

1.1 Power Consumption with a Battery Supply

Throughout this work, "low-power" is frequently listed as a design goal in battery-operated circuitry, and it would remiss if it was not given a contextual definition. "Low-power" in this work refers to circuitry that utilizes sub-threshold-level current biases which are largely below 1μA currents. In battery-powered sensing applications, a net sub-100μA current draw is a good rule-of-thumb benchmark for low-power circuits; however, this is highly dependent upon the complexity of circuits being used and application-space.

This work focuses on the sensing circuitry aspect of the wireless sensing nodes and does not focus on design improvements of the wireless microcontroller units (MCUs). However, MCUs can be the largest component of the power budget when not in a lower-power sleep-state mode. For example, current wireless COTS MCUs Particle Xenon and TelosB motes can consume up to 20mA and 24.8mA, respectively — not including any current draw from notification LEDs — with the RF transceiver active and the microprocessor at full capabilities [9, 10]. But, both boast sleep-states (i.e. transcievers off and microprocessor in a low-power, inhibited stated) with respective power consumptions on the order of μA draw (unmeasured by Particle at the time of writing) and 6.1μA for the TelosB platform. Therefore, it's advantageous for a paradigm change from how microcontrollers are traditionally employed to conserve a finite battery supply.

The paradigm change methodology is to (1) keep the wireless microcontroller in a sleep-state for as long as possible and (2) offload computation to an always-on, low-power front-end. The motivation is that most, if not, all computation can be done at a lower power consumption by the front-end circuitry than by an active wireless MCU. The MCU can be woken-up for certain digital computations and information transmissions. The amount of time the MCU stays in sleep-state is ultimately dictated on a per-application basis.

The front-end circuitry can be fulfilled by employing ASICs or by the aforementioned FPAA. It's useful to put into context the theoretical battery life for both circuits in an application example like thermistor temperature sensing application. The following example is for a hypothetical agricultural application for in-the-field frost warning system. Let's assume both types of sensing circuits leverage the same MCU and its sleep state time; this negates any advantage with respect to the choice of microcontroller. A thermistor's output of resistance is converted to voltage when a current is dropped across the sensor like that of a Wheatstone bridge circuit — See Figure 2.3 and Appendix A. Since there are millivolt changes (i.e. relatively small) at DC-like frequencies, instrumentation amplifiers like the Microchip MCP6N16 are a good candidate for small-signal changes, and the MCP6N16, in particular, can operate on a single supply voltage under 5V.

The MCP6N16 draws a quiescent current of 1.1mA, while a sensing circuit of Figure 2.3 can be completely biased at less than 500nA. When accounting for additional overheads of the FPAA, the net current draw for FPAA synthesized circuit like that of Figure 2.3 would most conservatively be 5μA. Furthermore, accounting for the resistive network power consumption, we can arbitrarily design for branch current of 50μA total when utilizing a Panasonic ERT-

Table 1.1: Operational lifetime using the ASIC front-end and on a single 4000mAh battery.

	ASIC Current	μProc. Current w/ Radio Off	μProc. Current w/ Radio On	Average Current	Battery Size(mAh)	Battery Efficiency	Operational Days
Active State: 1 Sec/Min	1.15mA	1.8mA	24.5mA	8.63mA	4000	80%	360
Sleep State: 59 Sec/Min	500nA	10μA	10μA	10.3μA			

Table 1.2: Operational lifetime using the FPAA front-end and on a single 4000mAh battery.

	FPAA Current	μProc. Current w/ Radio Off	μProc. Current w/ Radio On	Average Current	Battery Size(mAh)	Battery Efficiency	Operational Days
Active State: 1 Sec/Min	55μA	1.8mA	24.5mA	7.53mA	4000	80%	409
Sleep State: 59 Sec/Min	500nA	10μA	10μA	10.5μA			

J1VV154H 220$k\Omega$ thermistor. We can then compute the total ideal operational time if a battery's Ampere Hour rating is known. The tables 1.1 and 1.2 show the differences in operational lifetime using the ASIC and FPAA circuits on a single 4000mAh battery.

These calculations were based on a couple of assumptions. First, the MCUs will toggle the supply to the resistive network for the ASIC and the FPAA to sample the voltage divider. This reduces the parasitic current draw from the resistive network. The second assumption was that the ASIC and FPAA contain a comparator that would pulse HIGH if the temperature met some temperature threshold that was representative of a frost warning (while ignoring the dew point for this demonstrative example). This HIGH pulse would initiate the MCU to transmit a warning message. An additional assumption was this condition would occur 1/4 time of the year, which is represented by the metrics of the third column. The rest of the year would operate in the second column state with the radio off. The FPAA-based circuit would net 49 additional operational days compared to the ASIC circuit.

The 49 additional operational days can be attributed to an order of magnitude less current draw compared to ASIC front-end. Ultimately, there could have been a more extensive separation between the two designs if there was not a parasitic resistive network at the front-end and it was allowed to run 100% of the time. Chapter 7 demonstrates application examples where the MCU is asynchronously woken up. The result is a net magnitude of current difference from traditional periodic methods that can have an even larger effect on battery life compared to the demonstrative example of this chapter.

1.2 Outline of Work

The remainder of this document is devoted to elaborating on the various topics that have been previously enumerated in this introduction. Chapter 2 exhibits a complete wireless sensing system from its application, design, and deployment. In Chapter 3, I describe a high-side load switch used for FG programming that could also be employed for energy-harvesting circuits. Following in Chapter 4, I present an FG temperature compensation scheme for FPAAs. Chapter 5 gives a floating-gate transistor overview and analysis including their operation and programming which will be indispensable for the following chapters. Chapter 6 describes an FG below-ground programming array presented as a lower overhead alternative to many traditional works that utilize above-ground programming. The penultimate chapter exhibits a complete analog front-end for use with an ADC along with its use in a diverse set of end-use applications. The final chapter will serve as my conclusion and remarks on future work.

Chapter 2

Smart Refractory Sensor Systems for Wireless Monitoring of Temperature in Slagging Gasifiers

This chapter establishes the motivational basis for this dissertation work. A complete wireless sensor network is introduced to demonstrate the various components and their functional relationships. Furthermore, the end-use application of temperature sensing is a common sensing mode for IoT. The work completed in this chapter illustrates some of the hindrances of developing application-specific circuity which ultimately becomes the motivation for reconfigurable circuitry. This work was funded by the Department of Energy's National Energy Technology Laboratory Grant no. DE-FE0012383.

2.1 "Smart" Foundry Bricks

The premise for this Department of Energy (DOE) project was to develop "smart bricks" for a coal gasifier furnace to monitor the internal furnace conditions. The internal furnace environment experiences temperatures up to 1450°C and pressure up to 1000 psi for a duration over months. These extreme conditions make it a barrier to monitor internal furnace conditions which makes it difficult to determine it's operation. Currently, the furnace is treated like a black box where it is operated until the structural integrity is compromised such that it is unsafe to operate. The structure is then completely dismantled and rebuilt which is a lengthy expensive process.

However, if conditions within the furnace could be monitored, this information could be fed-back to operators (computers or human) so that they could modulate operational settings to moderate the internal conditions. The scope of the project would be to place a number of these "smart bricks" throughout the whole furnace structure to create a three-dimensional model for it's state of health. Presently, a single compromised brick is what ultimately leads to the catastrophic failure of the whole furnace. The three-dimensional model allows operators to find "hot spots" — harsh conditions in the furnace that could indicate impending brick failure at that particular point — and moderate furnace operation accordingly.

The "smart bricks" compose of the same refractory material that make-up the construction of the furnace, but have a lab-designed sensors embedded in them at the time of the brick casting. The terminals of the sensor exit the brick from the opposing side that will be exposed to the internal furnace. The sensor can then be interfaced with electronics to sense the internal conditions. The sensors that were developed by mechanical and chemical engineering team, and used to sense temperature, stress/strain and the internal furnace liner health. A cross-section of the furnace wall with a smart brick is shown in Fig. 2.1a. Fig. 2.1b shows the proposed testing furnace with the tentative sensor positions. The critical temperatures to be sensed by the circuit should be $1100° - 1400°$C as defined by the project proposal.

The sensing circuitry will be wireless and will run off battery power because there will be a number of sensors that could potentially be set in hard-to-access places on the furnace. The the test setup of Fig. 2.1b is for demonstration of the concept, but one can imagine the wireless sensing nodes encompassing the whole furnace. All the measured data at the sensor is transmitted to a centralized base-station node that listens for the data. The received data can then be processed and reviewed by an operator in real time.

2.2 Resistive Sensor

The two modes of temperature sensing that were explored in this project were voltage and resistance. The focus of this section is to examine the resistive sensors and its associated electronics. While the project explored both modes, the resistive sensors ultimately became the project's preferred avenue for temperature sensing.

A thermistor was chosen as the resistive temperature sensor and was designed by the

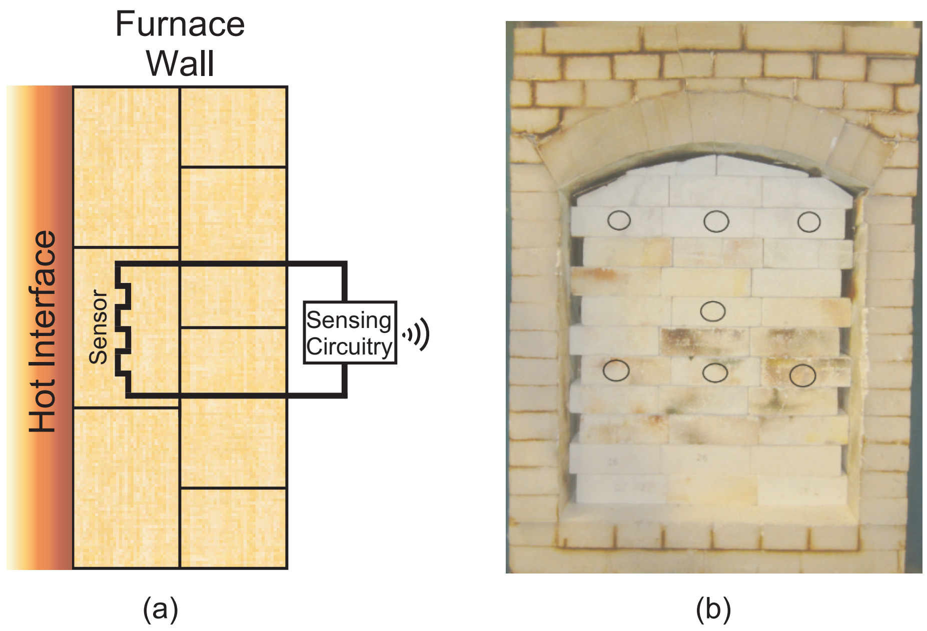


Figure 2.1: (a) Cross-sectional diagram of the smart brick furnace and (b) a photo of the proposed testing furnace with superimposed circles designating the proposed smart brick positions.

material sciences group. A thermistor is a resistor whose resistance is a function of temperature. There are two different types of thermistors: negative temperature coefficient (NTC) and positive temperature coefficient (PTC). An NTC thermistor's resistance decreases with an increase temperature, while the PTC resistance has the opposite behavior with an increase in resistance. The thermistors for this project are of NTC type due to their chemical composition.

An example thermistor used in this project is pictured in Fig. 2.2. The figure is a characterization plot of the measured thermistor resistance versus the temperature that the sensor experiences in the furnace. This characterization experiment is function of time since the furnace instantaneously heat-up or cool-down to the target temperature. Additionally the sensor is embedded in the furnace brick since this is how the sensor will be deployed in actual application. The first ten hours of the experiment has the furnace heating up to its highest temperature of 1350°C; this is shown as the red line utilizing the right y-axis. Within that time frame, we see the measured resistance (in blue utilizing the left y-axis) conversely decreasing in resistance during the same time period. This confirms the sensor is an NTC thermistor. The experiment holds the temperature of the furnace for a few hours before plateauing at lower successive temperatures. This is shown from 10 hours to 40 hours

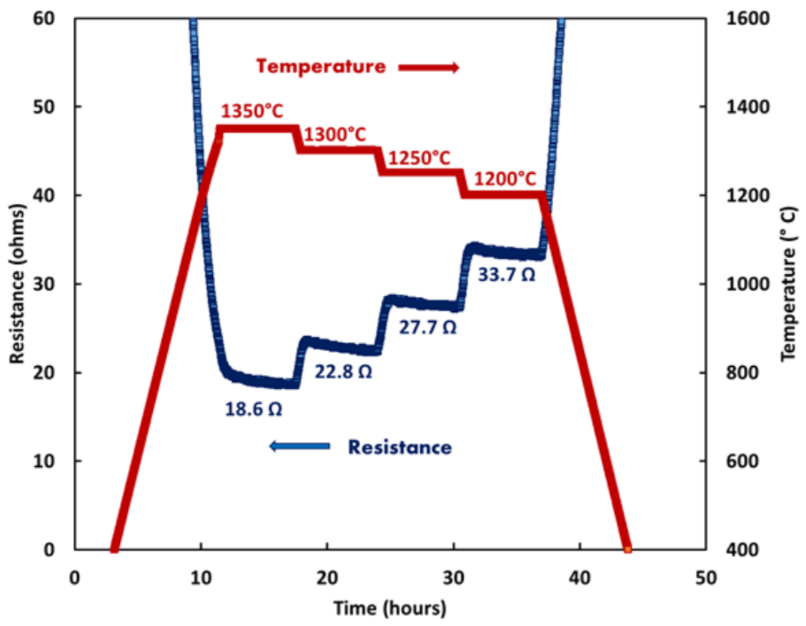


Figure 2.2: Experimental measurements of a thermistor characterization.

during the experiment, followed by a ramping down of temperature.

Designing a circuit for the thermistor in Fig. 2.2, we must account for the range of resistances the thermistor can take on while balancing sensitivity of the sensing circuit. The critical temperatures we want to be able to sense are temperatures above 1150°C. However, the range of resistances that correspond to the desired temperatures must be set within reasonable range to be useful. Using Fig. 2.2 as an example, the sensing electronics could be configured to sense a resistance between 15Ω and 115Ω. However, referring back to Fig. 2.2, if the temperature changed from 1200°C to 1300°C, and consequently the thermistor resistance would change from 33.7Ω to 22.8Ω, the sensing circuitry's output would only change about 10%. This example demonstrates that for the desired temperature range, there is relatively little change (only about 10%) in the output. On the other hand, every fabricated sensor will have mismatch among their resistance readings at the same corresponding temperature. As a result, the design must accommodate these 10-15% tolerances in their respective resistances due to non-idealities in the fabrication process.

2.3 Resistive Sensing Circuit

As described in the introductory section, the sensing circuitry needs to be battery-powered. Therefore, it needs to be energy efficient so that the batteries do not need to be replaced often – at least on the scale of months. The specific battery circumstances of this design will be three AA batteries as this supply will be shared with more than just the sensing circuitry. This constraint is set by the wireless microcontroller that will transmit the data and will be described in a following section. Furthermore, battery operation also means the circuitry needs to work on a single supply voltage.

Since the constraints are known and the solution needs to be low-power, the number of active components should be minimized. Furthermore, looking back at Fig. 2.2, we will need the circuitry capable of sensing a very small resistance whose change over the desired temperature range is in tens of Ohms. Measuring a resistance of this magnitude by means of Ohm's law (i.e. $V = I \times R$) will require either a robust voltage or current relative to the demands of low-power sensing. Since this design is operating off of a battery supply, we are limited to a supply voltage of 2.5V – this voltage is dictated by the microcontroller [11]. This supply voltage constraint means that the current will need to be in the milliampere range given the sensor change is in tens of Ohms resistance.0

Given that the overall constraints are known, we employed a resistive sensing circuit pictured in Fig. 2.3 [12], [13]. This design utilizes a traditional Wheatstone bridge circuit and only two operational amplifiers (op-amps). In its essence, this circuit senses a change in resistance, and in response, outputs a corresponding voltage. This output voltage can then measured by the wireless mote's analog-to-digital converter (ADC). The ADC's conversion represents this data as a codeword that can transmitted back to the base-station mote.

This design has a number of advantages for battery-powered sensing. The first is that only two op-amps are being used as opposed to an instrumentation amplifier, which in their most conventional forms, utilize three op-amps. The second is that this design can be easily designed for at the board level and integrated level. Third, this circuit can operate from a single supply rail if the operational amplifiers are chosen/designed correctly.

The qualitative operation can be understood and help in the analysis of the output voltage expression. Ignoring the second op-amp (OP_2) circuit for the time being leaves the Wheatstone bridge and op-amp one (OP_1). R_1 and R of the Wheatstone bridge are static resistors, while $R(1 + \delta)$ is the thermistor. The thermistor is placed in a negative feedback

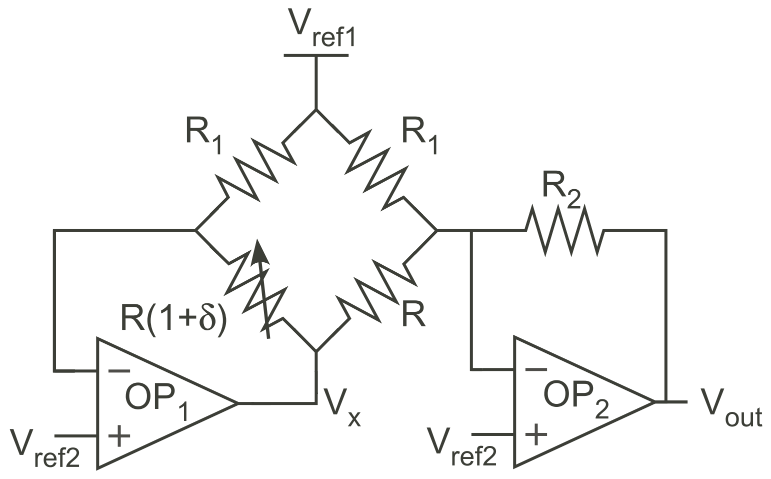


Figure 2.3: Thermistor sensing circuit that utilizes two operational amplifiers and Wheatstone bridge.

configuration of OP_1. Ignoring the right branch of the Wheatstone bridge simply yields a voltage divider down from V_{ref1}. This configuration permits a current to flow down from V_{ref1} to V_x, and the negative feedback configuration will attempt to maintain a consistent voltage drop across the thermistor such that its inverting input is the same value as its non-inverting input V_{ref2}. Stated another way, OP_1 will modulate V_x such that the same current is always flowing through the left branch of the Wheatstone bridge. OP_1 will only module V_x in response to a changing $R(1+\delta)$ resistance, which in our application will be when furnace temperature changes.

This leaves the right branch of the Wheatstone bridge and the inverting configuration of OP_2 . The right branch of the Wheatstone bridge is simply a voltage divider with static resistances R_1 and R. When the thermistor changes, and consequently V_x, the current through the right branch will change. Simply looking at the right branch and ignoring the OP_2 inverting amplifier, the net effect is that the voltage division seen at the node of R_1 and R will be V_{ref2} when $\delta \neq 0$. This voltage can be leveraged to produce an linearized output voltage with the help of OP_2's feedback configuration.

Taking the closed-loop OP_2 amplifier configuration into consideration, OP_2 will respond to the voltage difference between its two input terminals. If there is too much current running through the right branch (i.e. the voltage at the top of R is too high), OP_2 will lower V_{out} to draw some current out of the branch to maintain V_{ref2} at the non-inverting terminal. If there is too little current flowing through the right branch, OP_2 it will attempt to source the current difference by increasing V_{out}.

Using Ohm's Law and the ideal op-amp characteristics, the output voltage of the Wheatstone bridge sensing circuit can be expressed as the following equation. The full derivation is shown in Appendix A.

$$V_{out} = V_{ref2} + \frac{R_2}{R_1}\delta(V_{ref1} - V_{ref2}) \tag{2.1}$$

δ is the change in the sensing resistor (specifically, a thermistor). Also note that the output is not dependent upon the sensing resistance R (i.e. the resistance without any subscript), but is a function of its change. However, when designing and selecting an appropriate value for R in the right leg of the Wheatstone bridge, you must know the top and bottom range of $R(1+\delta)$. Ultimately, R is selected to be the middle value of the $R(1+\delta)$ range.

After R has been determined, one can select R_1 to determine an appropriate amount of current through the Wheatstone bridge legs. This would be dependent upon R. For example, if R is in tens or low-hundreds of ohms, the current through the Wheatstone bridge legs should be in the low milliampere range to create a large enough voltage drop across $R(1+\delta)$ for OP_1. This would also depend upon V_{ref1} and V_{ref2}. For our designs, V_{ref1} and V_{ref2} have been chosen to be V_{dd} and mid-rail, respectively. And finally, R_2 is selected to gain up the change in resistance to correspond the $R(1+\delta)$ range to an output voltage between $0V$ and V_{dd}.

2.4 Circuit Implementation

2.4.1 Voltage Regulator and Battery

The focus of this work has been to design a low-power sensing circuit for a wireless microcontroller, which implicitly means the power source is battery. A voltage regulator is needed to generate a stable and known voltage from an ever-changing battery output. Without regulation, the effective supply voltage would lower (non-linearly) over time. More specifically, the output voltage expression of relies on assumption of stable voltage references of V_{ref1} and V_{ref2}, which need to be held constant as the batteries' output voltage lowers over time. While it has not been explicitly stated or shown in the schematics up to this point, a voltage regulator is called for in this design.

The dynamic between the battery supply and the voltage regulator is for the battery to supply a larger than needed voltage and for the regulator to regulate the supplied battery voltage down to a desired value. Voltage regulators are able accomplish this functionality through a negative feedback configuration. For the case of this work, the main specification of the voltage regulator is to maintain a regulated 2.5V as the reference voltage and power supply of the sensing circuit. The output of three AA batteries is typically measured to about 4V, which gives plenty of headroom for the voltage dropout.

The voltage dropout specification is main reason for not choosing two AA batteries in series as the supply. The difference in measured output voltage of a two, full-capacity AA battery supply and the regulated voltage is on the order of 250mV. This is close to the minimum dropout voltage required for regulators of this application. If the two AA battery supply was employed, it would only be a matter of hours to draw down the capacity below the minimum dropout voltage. The issue is resolved by simply adding a third AA cell in series with the aforementioned two AA cell. This design choice does not affect any space and weight requirements of the project, and has the additional benefit of extended operation time. Ultimately, we chose a commercially available off-the-shelf component MCP 1700 as our voltage regulator [14].

2.4.2 Brick Characterization and Circuit Components

The smart bricks that were cast have been characterized and are shown in Fig. 2.4. This figure is a slightly different viewpoint from Figure 2.2 in that time has been removed from the figure, but ultimately conveys the same information. There is some variance among the smart bricks in that for the same furnace temperature, each respective thermistor does not map to the same measured resistance. As discussed in the previous section, the sensing range of $R(1+\delta)$ should encompass the top-end of the highest resistive sensor and the bottom-end of the lowest resistive sensor in the desired temperature range. Furthermore, only four of the bricks are being deployed for the final demo: Brick D, Brick C, Brick 3, and Brick 2. This means that the expected $R(1+\delta)$ sensing range to design the circuit around is 35Ω and 115Ω.

Knowing the data from Fig. 2.2 and considering the aforementioned design process of the circuit, the following values were selected for the various reference voltages and resistors of the Wheatstone bridge circuit in Table 2.1.

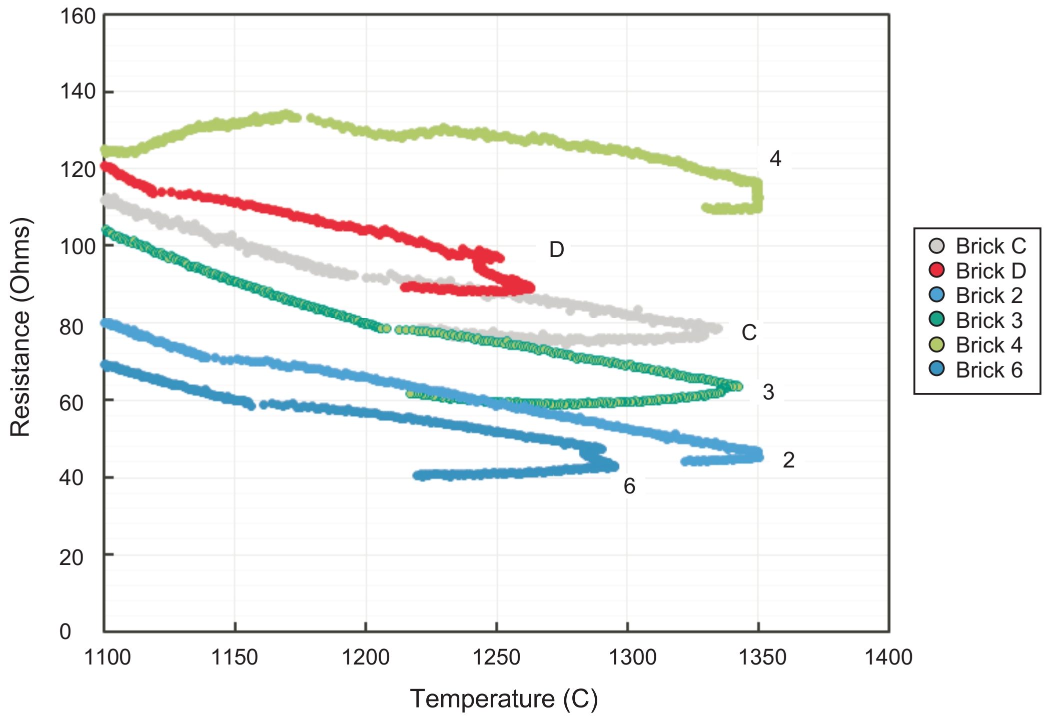


Figure 2.4: Six smart brick characterizations.

Fig. 2.5 allows one to map the output voltage of the sensing circuit to a specific temperature. The data of the figure is from two separate experiments layered over each other. The sensing circuit was physically measured with $R(1+\delta)$ changing in 10Ω increments, and the resistance measurement of the smart brick was measured during a characterization run. It is evident that the chosen sensor material has a nonlinear dependence on temperature, but the sensing circuit output is linear with the exception of when the op-amp does not operate in it's linear region at the supply-rail extremes. The nonlinear temperature dependence was expected for the sensor in this type of high-temperature, high-pressure furnace environment.

The wireless sensing circuit that will be used in proof-of-concept demonstration is shown in Fig. 2.6. It consists of a vertical stack of battery-pack, wireless mote, and sensing circuit from bottom to top. The sensing circuit contains two standard banana adapters to directly interface with the smart brick. Fig. 2.7 demonstrates how the wireless sensing circuit would be interfaced at a foundry furnace wall. The mechanical design of the wireless sensing circuit allows them to be vertically stacked. The top metal spacer can be screwed into the bottom of the taller spacer of another wireless sensing circuit.

The data shown Fig. 2.8 is the wireless sensor mote's data received at the master mote

V_{ref1}	$2.5V$
V_{ref2}	$1.25V$
R_1	680Ω
R_2	$1.133k\Omega$
R	75Ω

Table 2.1: Designed values for Wheatstone bridge resistive sensing circuit.

node. This figure demonstrates how one of the wireless sensing nodes fits within the whole system. The chain of events begins at the sensor. The sensing circuit amplifies and conditions the sensor signal for the mote ADC to measure. The mote takes the ADC measurement, encodes it into a code-word, and wirelessly transmits the measurement to the master mote. The master mote is the central location where all the measured data is aggregated and provides the human interface to the system. The master mote can then decode the code-word and plot the data.

The plot pictured in Fig. 2.8 is data from an experiment ran with a wireless sensing node on a WVU-made thermistor. The plotted data is the received data at the master mote that has been decoded from code-words into corresponding voltages. The data was measured from the sensing circuit and a smart brick from the same set of bricks pictured in Fig. 2.7. The experiment holds the brick at 1200°C for 40 hours and the data (one sample per minute) is received by the master node from the sensing node. One can use a look-up-table to map the decoded voltage back to a specific temperature, and in fact, looking back at Fig. 2.5, the voltage matches up well to the expected hot-junction furnace temperature. The change in the voltage output when the furnace is“held” at $1200°C$ reflects the realities of keeping a stable hot-junction temperature of $1200°C$ in a furnace. The feedback controls of the furnace continually keep a heats and cools which these changes are reflected in the measured data.

2.5 Conclusion

A wireless sensing circuit has been presented for a Department of Energy project. The project consists of lab-made temperature sensors embedded in refractory material that can withstand the high-temperature, high-pressure of a coal gasification furnace. It was demon-

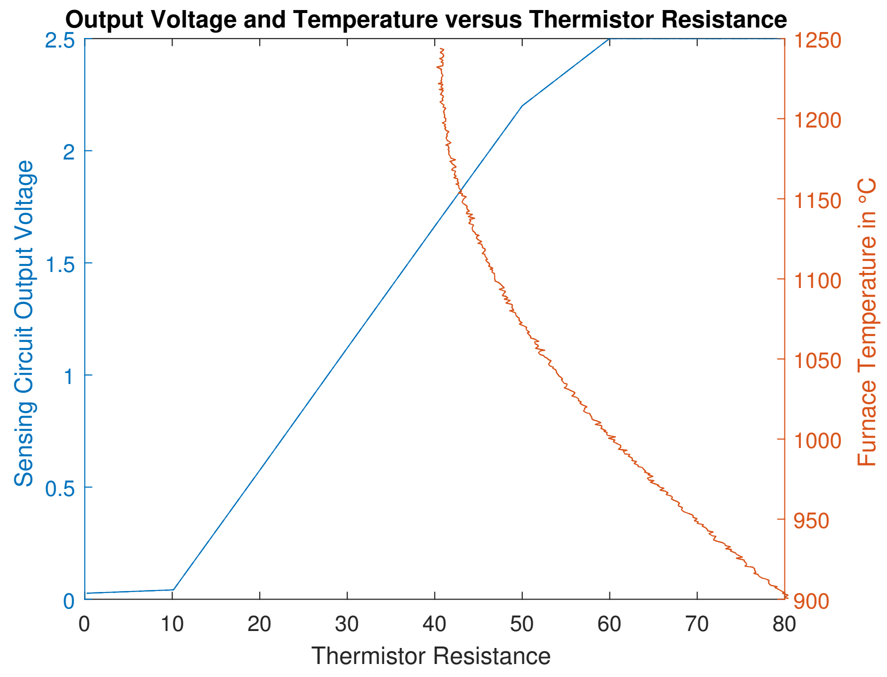


Figure 2.5: Left y-axis shows the sensing circuit output versus thermistor resistance. Right y-axis shows the hot-junction furnace temperature versus resistance. The figure is used to map output voltage to temperature.

strated that the sensors can be used to electrically monitor conditions inside the furnace. The focus of this work gives the design of electronic interface circuitry that is capable of measuring the sensor's signal (resistive), logging, and transceiving the data. The sensing circuit design consists of a low-power two op-amp design that is capable of running off of battery supply.

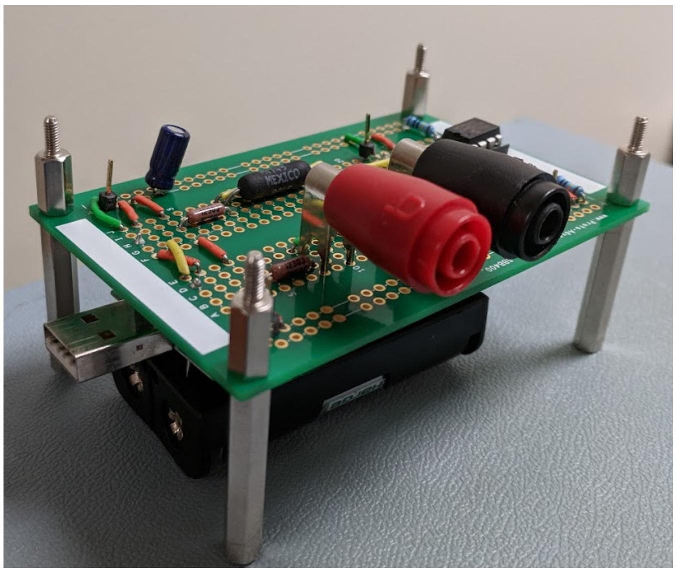


Figure 2.6: Physical design of the sensing circuit consists of battery, wireless mote, and Wheatstone bridge circuit stack from bottom to top.

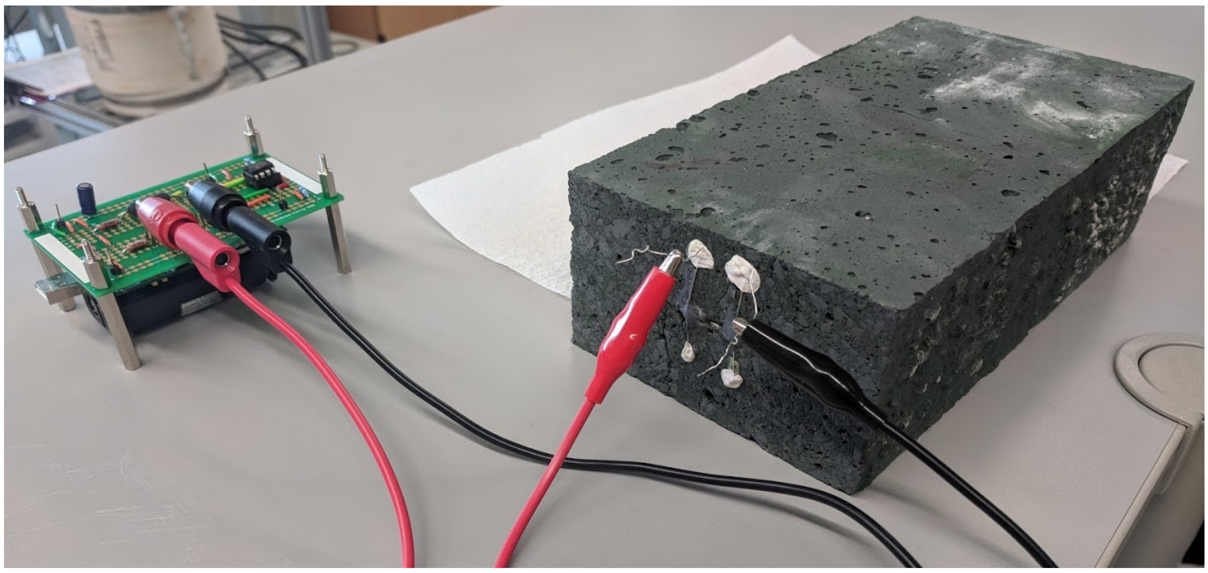


Figure 2.7: The sensing circuit interfaced with a smart brick. This a staged picture to simply demonstrate how the interface connection would look like in a real deployment.

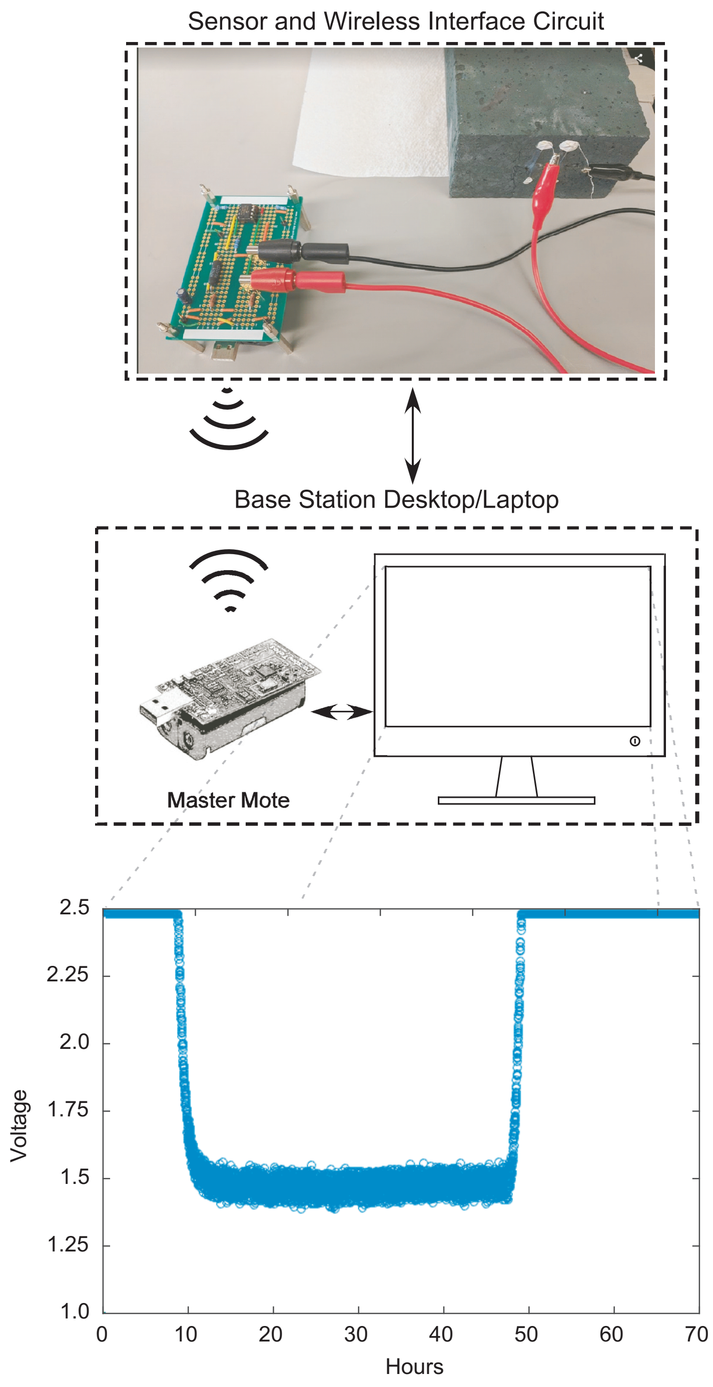

Sensor and Wireless Interface Circuit
Base Station Desktop/Laptop
Master Mote
Voltage
2.5
2.25
2.0
1.75
1.5
1.25
1.0
0
10
20
30
40
50
60
70
Hours

Figure 2.8: Demonstration of the wireless sensing node and its fit within the smart brick sensing system.

Chapter 3

High-Side Switch

Reconfigurable analog circuitry has many barriers to entry and there is a large need for lowering their peripheral-circuit overhead to encourage wider use. These peripheral circuits assist in one of FGs two operational modes: Run-mode and program mode. To that end, this chapter presents work that supports the former responsibility but is presented here because it has larger applications in the power-management and energy harvesting space. This fits our larger narrative of advocating for low-power analog circuitry but also demonstrates that more can come of FG development beyond its immediate usage.

3.1 High-Side Switches

As low-power electronics become more prevalent for IoT applications, power management is a significant concern. In low-power systems, multiple voltage sources are often used to provide supplemental energy or to reduce power consumption. Examples include switching the supply voltage from a battery to an energy-harvesting system or switching unused subsystems to a lower supply voltage. Selecting a specific voltage source to use at any given time requires a high-side load switch, which is a switch that can connect/disconnect a supply voltage to a particular load.

The conventional usage of a high-side load switch (HSS) is demonstrated in Fig. 3.1a. The load is connected/disconnected from a single supply voltage with a digital selection signal denoted by V_{sel}. A real-world power management example would be a smartphone needing to shut-off the power supply to a auxiliary component like a bluetooth transceiver. High-side switches used for these purposes are well known and readily available components [15, 16].

A slightly more complicated configuration of a HSS is shown in Fig. 3.1b, where a circuit's supply voltage (V_{supply}) is multiplexed between two voltage sources. This configuration would be beneficial in energy harvesting applications for conserving the supply power of the battery by switching to an alternative energy source like solar or thermoelectric. These kinds of products have began to emerge in the past couple of years [17].

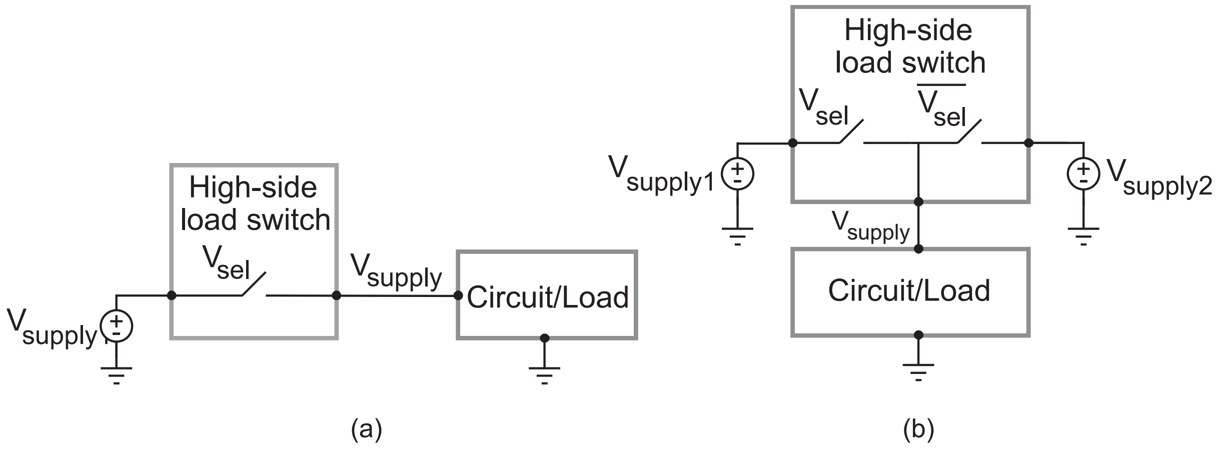


Figure 3.1: (a) Block diagram of a conventional high-side load switch and (b) a multiplexing high-side load switch.

The motivation for this research is to present a HSS suitable for floating-gate (FG) operations, which requires multiplexing two supply voltages to the FG transistor cell. FGs are explained in a more exhaustive manner in Chapter 5, but they are most well known for flash memory applications in CMOS technologies. FGs have two modes of operation: run and programming. These two modes correspond to two entirely different supply voltages. Fundamentally, this is different from the aforementioned energy harvesting example in that from the load's perspective, it perceives the same supply voltage regardless of energy source. Therefore, one of the main design criteria of a FG HSS is the capability to multiplex two different supply voltages like that of Fig. 3.1b.

There are a number of reasons why there are no existing circuits for this application. First, the FG programming voltage is higher than the rated voltage of the process, so the design must be able to accommodate voltages higher than the rated voltage. Secondly, the HSS design cannot depend on the programming voltage to always be present. Therefore, a HSS design cannot be made with the assumption that since the programming voltage will be the highest of the two voltages and should act as the supply voltage for the entire circuit. The implication of these two points is the HSS must (1) accommodate for the cases of two different voltages and (2) be capable of operation with the possibility of only one voltage being applied.

In a FG system, there are usually arrays of FGs for digital storage or analog computation. The FG HSS will act as the voltage supply to all FG cells. However, it would not be a stretch of the imagination to re-purpose this design to in carrying out the same operation as the energy-harvesting battery-life extender of [17]. In fact, there are a number of advantages of this design that will be detailed in the following section that include lower chip footprint, lower power consumption and operational insensitivity to supply voltage values and their changes.

The layout of this chapter is organized as follows. In the following section, an overview on current state of HSSs and the rationale for a new high-side switch is given. The final section is devoted to FGs. A historical context is given in terms of the development of both analog and digital applications in addition to programming a FG. This section will also serve to show the application overlap between traditional HSSs and FGs. The final section 3.3 presents our design of the FG HSS along with its mechanics and operation.

3.2 Power management and energy-harvesting systems

The driving force behind this research was originally for floating-gate programming, however, power management and energy harvesting systems could also directly benefit from this development of a new high-side switch. Before the case for this can be made clear, the backgrounds of HSSs are be presented to the reader. There are two main types of HSSs: high-side power switch and high-side load switch. The high-side *power* switch is capable disconnecting/connecting the supply voltage and regulating power consumption of a load. Examples of this type of HSS are prevalent in the automotive industry where there are many components with high voltages and large current draws that require power delivery regulation. On the other hand, the high-side *load* switch simply functions as a multiplexer, but does not limit power consumption. An excellent example of this type would be the previously given HSS whose function it was to turn on/off the bluetooth transceiver in a mobile device. For both types of HSS, their operation satisfies one of the objectives of power management to reduce power consumption.

Power management is also employed in energy-constrained applications. In battery-powered systems, multiple voltage sources are often used to provide supplemental energy or to reduce power consumption. Examples include switching the supply voltage from a battery to an energy-harvesting system (e.g. solar power) [17]. Note that the difference in this application setting is that the high-side switch multiplexes between two supply voltages as opposed a single supply and floating node. In this context of selecting a particular voltage source classifies it as a high-side *load* switch because there is no concern for current regulation to the load (e.g. a circuit).

There has not been as much of a focus on developing new high-side load switch designs. One of the reasons for this is that a high-side load switch is schematically realized with a single transistor which fulfills its operational constraints. Furthermore, most applications do not need to multiplex more than one supply voltage, so there are not many configurations to create with a single switch. Most attention has been on improving high-side power switches. These improvements are seen in energy efficiency since most of development is spent not on better switches, but on improving regulators and gate drivers.

Typically, switch design for both types of HSSs are comprised of NMOS devices. For an NMOS transistor to operate as a HSS, the gate-to-source voltage must be at least a threshold voltage above the drain-to-source voltage. The implication of this constraint is that the HSS design must include a gate driver capable of driving the gate voltage above the supply voltage. This ultimately means more support circuitry. The alternative is to use a PMOS transistor as the switch, which would obviate need of the gate driver and makes it the more natural choice in passing a supply voltage. This was ultimately the design choice taken for this HSS that will be presented in the next section. However, most designers opt for NMOS switches. The rationale for this choice is that PMOS transistors can be up to 3 times larger than NMOS transistors for the same switch on-resistance due to device mobility of charge carriers.

One of the most active research areas for HSSs is power management for industrial environments like the automotive industry. Newer vehicles are becoming more interconnected than before with sensors and control systems. These designs require the ability to withstand large voltages in the range of tens of volts (up to $55V$) and large current draws in the single amperes (up to $3A$) [18, 19]. The switch resistances for these designs range from about sub 1Ω to 10Ω. For these industrial applications, a HSS is usually designed in a Bipolar, Complementary MOS, Double-diffused MOS (BCD) technology. This allows for a fully-integrated and robust design with flexibility of technology choice. Additionally, these HSSs are of the power switch type since they must have short-circuit protection and high-voltage protection from demagnetization (due to inductive loads).

Since most applications that HSSs have been designed for have been high-voltage and high-current applications, there is a good argument to be made that the switch resistance should be as small as possible to minimize the voltage drop. On the other hand, for energy-constrained power management applications like energy-harvesting, there is less of a case to be made since the value of switch resistance can likely be softened; the loads usually use smaller supply voltages and draw smaller amounts of current (i.e. sub $1m$A), so voltage drop seen by the loads is not very drastic. This makes a compelling case for utilizing PMOS switches in energy-constrained applications considering a significant power savings may be made by not having a gate driver. This is one of the justifications for PMOS switches in the HSS that will be presented in the following section.

Our stated objective is to design a HSS capable of multiplexing two different supply voltages to a load. However, among the current HSS designs, there are not any that meet all of our desired criteria. On one extreme is the energy-harvesting battery-life extender of [17]. It was designed to multiplex a battery source and an ultra-low energy harvesting device like a thermopile whose voltage does not get above $1V$. On the other extreme are industrial HSSs that are designed to withstand supply voltages in tens of volts, but are realized in a Bipolar, Complementary MOS, Double-diffused MOS technology. This is not a viable technology for FGs because they are built in standard CMOS processes and are arrayed by the hundreds to millions on a single chip. And to the best of author's knowledge, there is not any published industrial HSS design that exhibits multiple supply voltage multiplexing. Furthermore, both of these HSS solutions consists of a complex system of components like boost converters, regulators, and off-chip passives (e.g. inductors). These components are not necessary for the needs of FG programming and consume extra power and a larger physical footprint. The requirements of the FG HSS need to be low-power (sub $10nW$) and as small as physically possible for given switch resistance.

3.3 High-Side Switch

The previous section focused on the background information of a HSS and the justification for a new voltage multiplexing HSS. This new type lends itself useful for FG programming and energy harvesting power management applications. In particular, a HSS with a low design overhead and the ability to multiplex voltages beyond the nominal supply voltage are two features unique to this design. Many of the current HSSs with similar features were not designed in a CMOS technology and contain unnecessary components for FG programming. The design choices to achieve these goals will be presented in this section.

In a CMOS process, either an n-channel or p-channel MOSFET (nFET or pFET) can be used as a switching element. Using a MOSFET as a switch means one is operating the transistor in the triode region because $V_{DS}<V_{GS}-V_{th}$. Furthermore, when operating a MOSFET as a closed switch in triode, it's a reasonable assumption that there will be a very low V_{DS} drop across the channel — in fact, most load transistors are designed such their ON-resistance is low, and therefore, have a very low voltage drop. Therefore, the triode equation for the drain current,

$$I_D = \mu_n C_{ox}\frac{W}{L}\left[(V_{GS}-V_{th})V_{DS}-\frac{1}{2}V_{DS}^2\right] \tag{3.1}$$

can effectively be reduced by approximating the $\frac{1}{2}V_{DS}^2 \approx 0$ which yields following expression:

$$I_D = \mu_n C_{ox}\frac{W}{L}\left[(V_{GS}-V_{th})V_{DS}\right] \tag{3.2}$$

where

μ_n	electron mobility constant for nFET;
C_{ox}	oxide capacitance;
$\frac{W}{L}$	FET device dimensions;
V_{th}	threshold voltage;
V_{DS}	drain-to-source voltage.

Operating in this extreme is referred to as the deep ohmic region because there is a linear relationship between the drain current I_D and voltage V_{DS}. With (3.2) we can characterize the effective MOSFET ON-resistance with dividing it by V_{DS}:

$$R_{DS_{ON}} = \frac{1}{\mu_n C_{ox}\frac{W}{L}\left[(V_{GS}-V_{th})\right]} \tag{3.3}$$

(3.3) shows that the designer of the MOSFET switch can maximize $\frac{W}{L}$ to lower the on-resistance. Furthermore, the designer has the choice between pFET and nFET which would change the charge carrier mobility constant. In (3.3) shows μ_n which designates this device as an nFET switch, where for a typical CMOS process $\mu_n \approx 3\mu_p$. Considering that the objective of a high-side load switch is to pass a supply voltage, a pFET is a more natural choice to act as the switch even though nFETs have a better $R_{DS_{ON}}$ per unit area because of its larger mobility of charge carriers. To achieve the same objective with nFET switches, a voltage step-up converter must be used to generate a gate voltage at least one threshold voltage above the supply voltage that is being passed; this added circuitry comes at the expense of greater silicon area, power consumption, and design complexity.

The conventional usage of a pFET-based high-side load switch is demonstrated in Fig. 3.2, where a load is connected/disconnected from a single supply voltage. The source and well of the single pFET are connected together to ensure that the well voltage is at the highest potential. However, this configuration poses a problem when there are multiple, differently valued, and dynamic supply voltages each coupled to the load via different pFETs. The well of each pFET switch must be connected to the highest voltage encountered, or else there is significant possibility that a well-to-diffusion junction will become forward biased, causing the high-side load switch to fail.

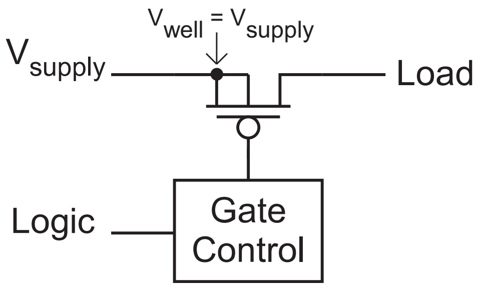


Figure 3.2: Block diagram of a conventional high-side load switch.

In this work, a pFET-based multiplexing high-side load switch that is capable of connecting multiple supply voltages to a single load is presented. To ensure that no junction is ever forward biased, a technique for dynamic biasing of the well potentials is employed. This high-side load switch is capable of passing supply voltages above the rated supply voltage for a process. As a result, this circuit is able to connect the large voltages required for hot-electron injection when programming FGs used in flash memory and analog non-volatile memory [8], which was the original application of this high-side load switch.

While recent work on high-side switches has largely focused on using specialized processes (e.g. Bipolar-CMOS-DMOS [19, 20]), this work is focused on utilizing standard devices in a CMOS process. In the following, we present the operation of the CMOS high-side load switch.

3.4 Circuit Implementation

When using pFET switching elements in a high-side load switch, the wells must be biased at or near the highest potential to prevent the possibility of forward biasing. Fig. 3.3 displays our multiplexing high-side load switch that incorporates pFET switches along with adaptive well biasing. The adaptive well biasing occurs through the well-selection transistor pair, M_5 and M_6, by passing the higher of the two supply voltages to the common well, V_{well}. This novel technique was re-purposed from [21] which adaptively biased the wells of pFET charge-pump stages. In most charge pump designs prior to [21], the stages were designed with nFET devices. As a consequence, when the charge-pump stages were cascaded, each successive stage beyond the first increasingly suffered from the body-effect which ultimately lowered the theoretical charge pump output voltage. While this same method solves two different issues for two different applications, it ultimately shows a pFET-type solution born out of issues from an nFET.

The HSS schematic of Fig. 3.3 can be compartmentalized into its operational parts. The transistors M_7 and M_8 are the switch elements that multiplex either $V_{supply1}$ or $V_{supply2}$ to the output, V_{supply_out}. $M_{5,6}$ senses and dynamically bias the common V_{well} connection among the pFETS. Transistors M_{1-4} form a standard level shifter driven by an inverter and its complement signal; M_{1-4} and the inverter perform the function of the gate-control block of Fig. 3.2. The inverter should be powered either by the lowest supply or chip V_{dd}.

The determination of which supply source is connected to the output load is specified by digital selection V_{select}. When V_{select} is HIGH, M_2 is turned ON and the gate of M_8 is pulled towards ground, thus connecting $V_{supply2}$ to V_{supply_out}. At the same time, M_3 is turned ON and level shifts the gate of M_7 to the highest potential (i.e. n-well potential) to turn OFF the unselected switch. Recall that this n-well potential is established by the dynamic biasing from the well-selection transistor pair ($M_{5,6}$) and will be connected to the higher of $V_{supply1,2}$ regardless of the value of V_{select}. When V_{select} is LOW, the same, but complementary operations will occur to turn ON M_7 and turn OFF M_8.

The well-selection pair ($M_{5,6}$) has a clearly defined output when $V_{supply1,2}$ are significantly different from each other; the common well has a potential equal to the maximum of $V_{supply1,2}$. However, when $V_{supply1,2}$ are within approximately one threshold voltage (V_T) of each other, $M_{5,6}$ turn OFF, and the well voltage floats. Traditionally, this well-selection pair has been used exclusively for when $V_{supply1,2}$ are significantly different (e.g. the charge pump of [21]).

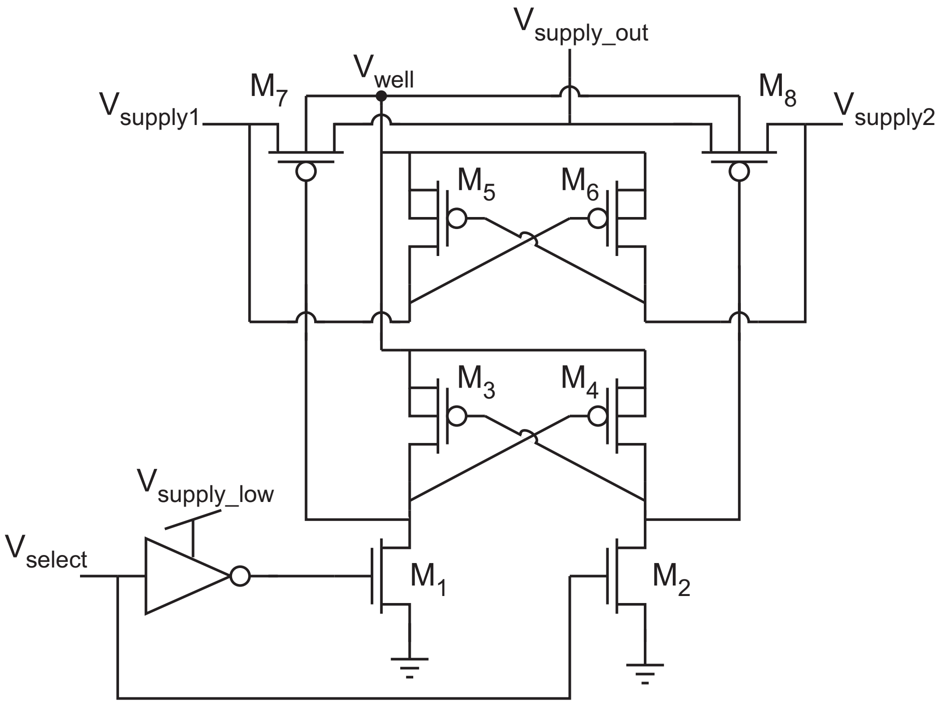


Figure 3.3: Schematic of the multiplexing high-side load switch using adaptive well biasing.

However, we will describe how the well-selection pair continues to keep the high-side load switch operating as desired, even when $V_{supply1,2}$ have similar values.

When $V_{supply1} \approx V_{supply2}$, V_{well} is floating but will stay in close proximity to $V_{supply1,2}$. For instance, V_{well} cannot float higher than $V_{supply1,2}+V_T$ since that would turn ON $M_{5,6}$ and pull V_{well} back towards $V_{supply1,2}$. Also, V_{well} cannot float too low, since that would forward-bias the diffusion-to-well p-n junctions of Fig. 3.4, thus adding charge to the floating well. As a result, V_{well} would begin pulled back towards $V_{supply1,2}$. In either scenario, V_{well} stays close enough to $V_{supply1,2}$ that there are no significant effects on the performance of the high-side load switch.

The exact value to which V_{well} floats depends on how quickly $V_{supply1,2}$ are changing. If $V_{supply1,2}$ are moving slowly (e.g. DC changes) in either the same or opposite directions, then V_{well} will settle to a value slightly less than V_{max}=max($V_{supply1,2}$); experimentally, we have found this voltage to be within 200mV of V_{max}. This DC experimental measurement of a buffered V_{well} node is shown in Fig. 3.5. The reason for this behavior is due to the reverse-bias leakage current from the well-to-substrate pulling V_{well} down and also a slight forward-biasing of the diffusion-to-well potential for V_{max}, thus establishing an equilibrium. These two different interfaces are represented in Fig. 3.4, where the parasitic pn-junctions

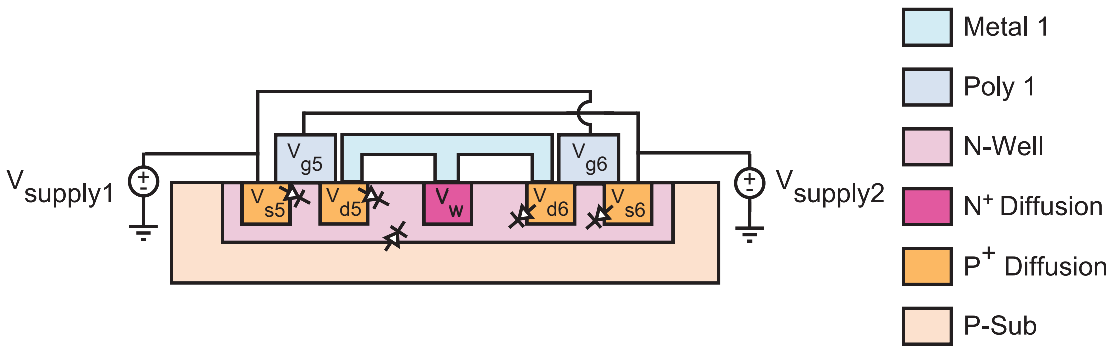


Figure 3.4: Physical cross-section of the well-selection transistor pair with schematically overlaid parasitic pn-junctions.

are overlaid schematically.

For faster changes in $V_{supply1,2}$, the well capacitance keeps V_{well} in close proximity to V_{max} until $|V_{supply1} - V_{supply2}| > V_T$ and V_{well} becomes strongly connected to the larger of $V_{supply1,2}$. The transient case is shown in Fig. 3.6 with V_{well} in the middle plot. V_{well} does not fall below $4.75V$ where $V_{supply1,2}$ meet.

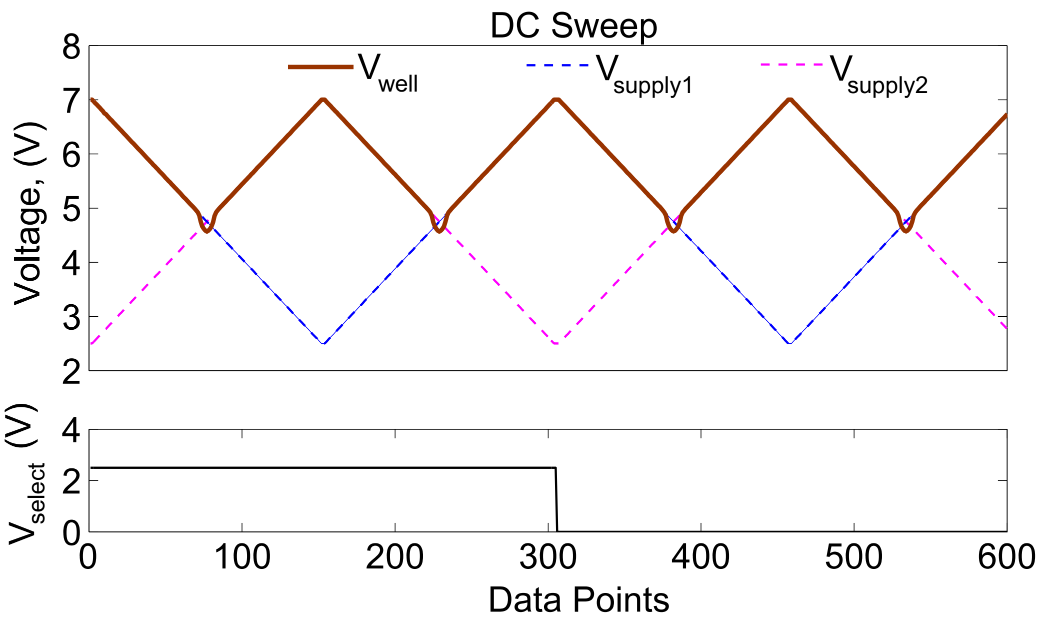


Figure 3.5: DC measured results showing V_{well} (brown) floats consistently for every combination of V_{select} (blue) $V_{supply1,2}$ (blue, pink) when $V_{supply1,2}$ are within a $200mV$ of each other.

3.5 Results

The high-side load switch of Fig. 3.3 was fabricated in a 0.35μm CMOS process using only standard 3.3V devices (i.e. no thick-oxide I/O devices). The measured results are displayed in Fig. 3.6, which varies both supplies over time while switching the selection signal. Fig. 3.6 covers a combination of various signal situations and shows the output, V_{supply_out}, following its selected supply voltage. The middle plot shows the waveform of the common well potential, which was measured on an auxiliary circuit since this node was too sensitive to pin out; this subplot shows that the well-selection transistor pair dynamically biases V_{well} to the highest supply voltage, and when $V_{supply1} \approx V_{supply2}$, V_{well} stays in close proximity among all switching cases. The bottom plot is the selection signal V_{select} that dictates which supply voltage is chosen to output to V_{supply_out}. The experimental results also demonstrate that the circuit can safely withstand temporary voltages that exceed the rated 3.3V supply voltage of the process. Care should be taken so that no device undergoes junction or oxide breakdown, which set the limit for V_{max}. Typically these breakdown mechanisms occur at 2-3 $\times$ V_{dd} in most processes [22]; our circuit was limited to $\approx$ 7V.

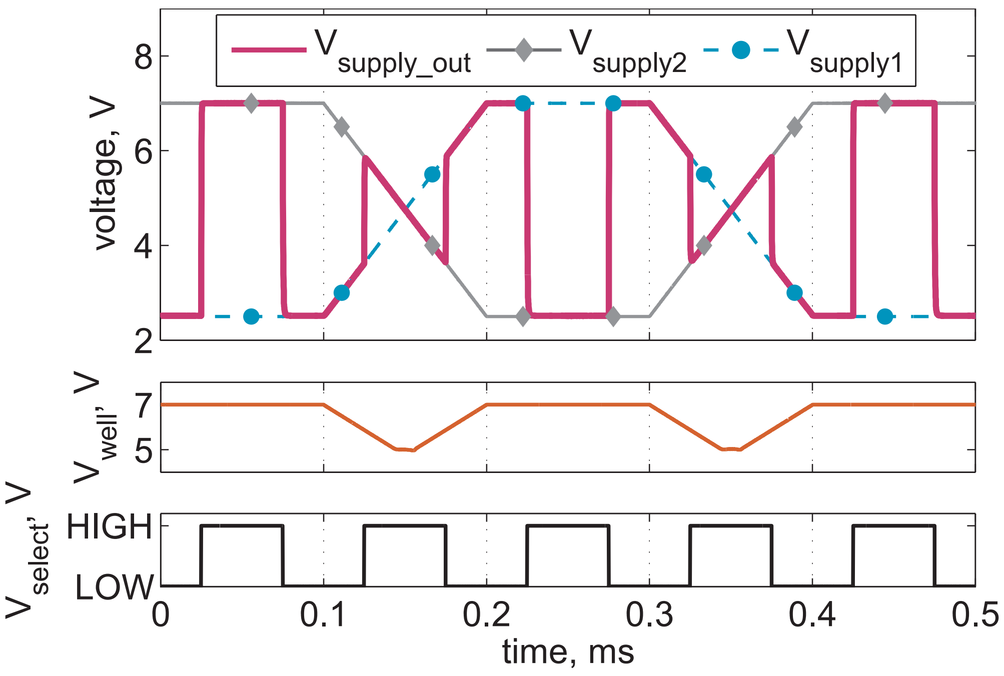


Figure 3.6: Measured transient results showing the (top) input/output behaviour, (middle) well potential, and (bottom) selection signal. Logic HIGH = V_{supply_low} and logic LOW = $0V$.

Operation beyond the rated 3.3V is especially useful for our target application of programming non-volatile analogue memory using floating-gate (FG) transistors [8]. For example, hot-electron injection is a common method of precise FG programming which requires large voltages outside of the typical supply rails to create a large source-to-drain potential [23]. In this 0.35μm process, $V_{sd} \approx 6.5V$ is required. Therefore, we temporarily enable a charge pump to generate this high voltage and use the high-side load switch to connect the FG device to the high voltage while programming. Otherwise, we connect the FG device to chip $V_{dd} = 2.5V$. The programming procedure is demonstrated in Fig. 3.7 where a charge pump is connected to $V_{supply2}$ and is enabled at $t = 1s$. Once the charge pump output is at $6.5V$, the high-side load switch connects V_{supply_out} (i.e. the FG transistor) to the charge-pump output ($V_{supply2}$) via V_{select}. When programming has been completed, the high-side load switch reconnects the FG transistor to chip V_{dd} ($V_{supply1}$), and then the charge pump is disabled and allowed to discharge slowly towards ground. A die photograph of the high-side switch and charge pump are displayed in Fig. 3.8.

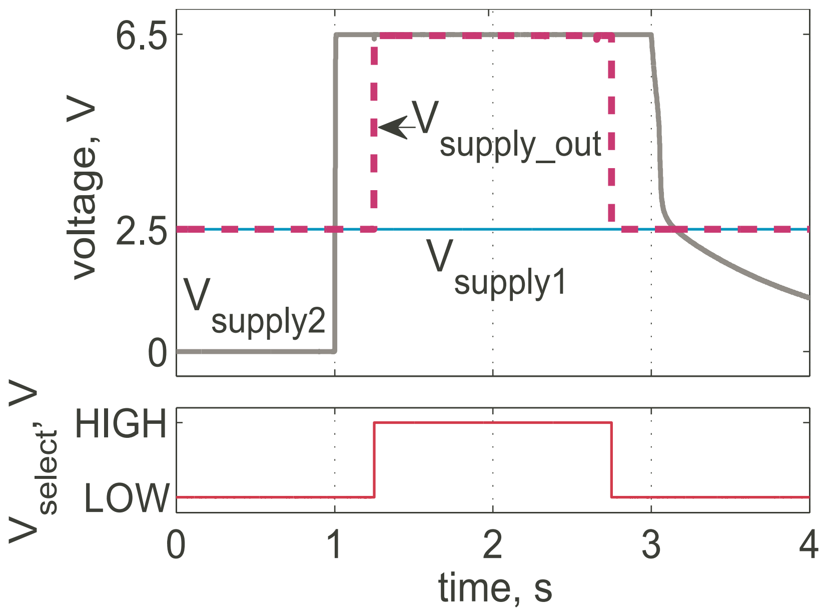


Figure 3.7: Floating-gate programming demonstration using the high-side load switch and charge pump where $V_{supply1}$, $V_{supply2}$, and V_{supply_out} is shown in teal, gray, and dashed red, respectively.

Measured characteristics of the multiplexing high-side load switch are displayed in Table 3.1. The ON-resistance of both pFET switches was measured to be 45Ω, as designed, to meet our constraint of $R_{ON} \leq 50\Omega$. The ON-resistance is adjustable by sizing the switching transistor dimensions ($M_{7,8}$) to meet an application's design constraints as was shown in

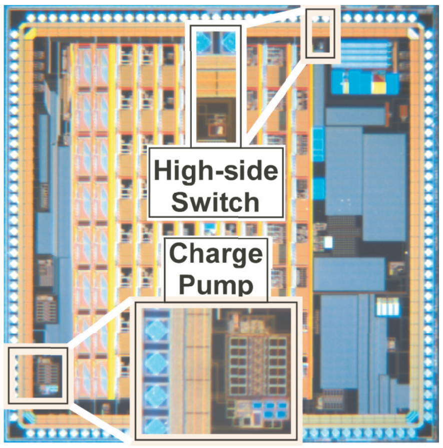


Figure 3.8: Die photo of chip with insets showing the detail of the charge pump and high-side switch.

(3.3). One additional consideration a designer should take into account are the parasitic resistances that occur in layout. For example, Fig. 3.9 shows the physical layout of a HSS with a designed ON-Resistance of 2Ω, where the device dimensional width was on the order of 125μm. However, the physical on-chip connections between pads and the pins of the 2Ω HSS contributes to the overall measured ON-resistance, which was the unfortunate case with a HSS that was fabricated in Fig 3.8. The 2Ω HSS had a die placement shown in Fig. 3.10 and the the parasitic resistances of the pad to pin connections contribute $\approx$ 100Ω. Thus the effective measured ON-resistance was approximately 102Ω. Our physical measurements were verified parasitic resistance simulation.

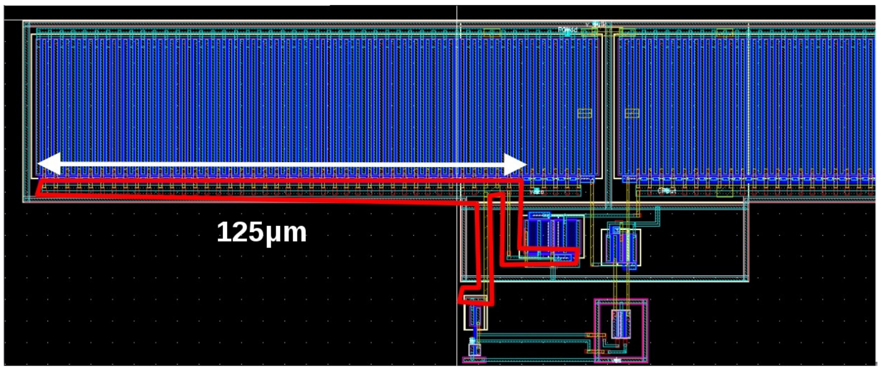


Figure 3.9: Physical layout of a 2Ω HSS. Outlined in red is a single connection or node, which is 125μm long.

Fig. 3.11 shows the measured power consumption of the circuit when sweeping $V_{supply2}$ and while holding $V_{supply1} = 2.5V$. The trend can be attributed to a combination of reverse-

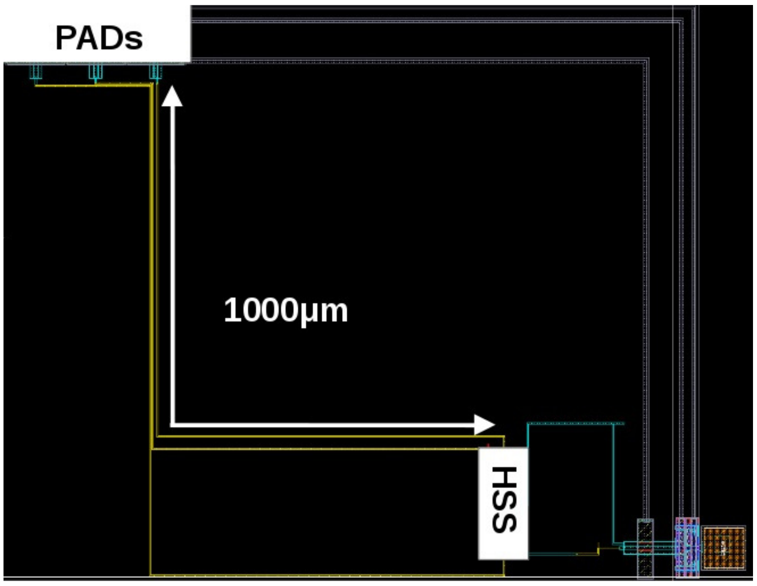


Figure 3.10: The physical connection between the HSS and the padframe of a fabricated design contributes non-negligible amount of parasitic resistance to the ON-resistance.

biased p-n junction current and subthreshold leakage current. As $V_{supply2}$ decreases, the total current consumption greatly decreases, thereby reducing the static power consumption. The static power consumption of $4.6nW$ shown in Table 3.1 was taken under the conditions of $V_{supply\_low} = V_{supply1} = 2.5V$ and $V_{supply2} = 7V$, which represents a worst-case power consumption scenario for this circuit. Note also from the right plot of Fig. 3.11 that there is a slight increase in static current when $|V_{supply1} - V_{supply2}| < V_T$. This current increase helps to validate our discussion regarding the operation of the well-selection pair ($M_{5,6}$) when $V_{supply1} \approx V_{supply2}$. When $|V_{supply1} - V_{supply2}| < V_T$, V_{well} is at a voltage between $V_{supply1,2}$. Thus, a slight forward biasing from the higher V_{supply} to the well causes current to flow into the circuit and is counteracted by a reverse-bias current flowing out of the circuit from the well to the lower V_{supply}.

Two points can be made about this phenomenon. First, note that the supplier of the current into the circuit is whomever is the higher V_{supply} and hence why we see this role change as $V_{supply2}$ is swept above $V_{supply1}$ at $2.5V$. Secondly, note how the currents counteract in equally but opposite amounts. They eventually self-bias as $|V_{supply1} - V_{supply2}| \approx V_T$. These currents are small (i.e. $< 300pA$) and do not impact the operation of the circuit when $V_{supply1} \approx V_{supply2}$.

Parameter	Measurement
ON-Resistance	45Ω
Rise time	$10.5ns$
Fall time	$16.5ns$
Turn-on delay time	$11.5ns$
Static power consumption	$4.6nW$
Dimensions	$67 \times 84\mu m$

Table 3.1: Measured Characteristics

3.6 Conclusion

A multiplexing high-side load switch with adaptive well-biasing was presented. It was fabricated in a standard $0.35\mu m$ CMOS process. Although the rated voltage is $3.3V$, the circuit is capable of operating at a much higher supply voltage. The load switch can serve a variety of applications, such as energy-harvesting power management and floating-gate programming. While this design meets all the criteria that was required for our applications, there could be improvements made to future iterations.

The design is schematically small ($\approx$ 10 transistors), so there is very little room for improvement in this aspect of it without a completely new schematic. Even considering the lower mobility of pFETs that contribute to larger sized switch devices, this design choice is validated in that the additional support circuitry physical dimensions and power consumption of charge-pumps and other circuitry is not needed for operation. However, the case could be made that there should be improvements made on the physical aspects of design to improve performance. The measured ON-resistance was in the tens of Ohms, which could be lowered to single Ohms. Recall that there are chip-level parasitic resistances that are not accounted for in the transistor-level design. Parasitic resistances may be reduced by increasing metal trace widths, placing the circuit closer to its connections, and employing parasitic extraction simulations.

In terms of applications, the circuit has been well-tested in floating-gate programming environment. However, its capabilities for energy-harvesting power management have only been simulated and tested in a contrived lab environment. While the circuit meets the conditions and criteria for energy-harvesting power management, it has not yet been used

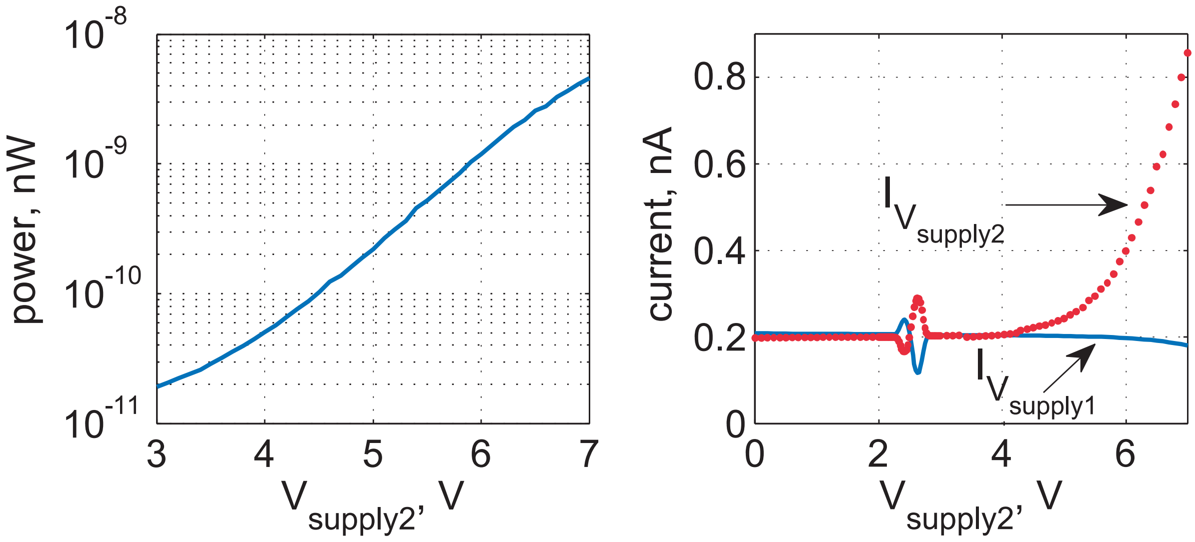


Figure 3.11: The left plot shows measured static power consumption as a function of the $V_{supply2}$ voltage. The right figure shows the respective measured current contributions by the two supply voltages as $V_{supply2}$ voltage is increased.

in a real application. For example, the HSS coupled with some energy harvesting circuitry would be an interesting complement to the work in high thermal energy environment of in Chapter 2.

Chapter 4

Temperature Compensation of Floating-Gate Transistors in Field-Programmable Analog Arrays

Post-fabrication reconfigurability is the feature that makes FGs attractive to analog designers. However, thinking about their real-world use, one quickly realizes that for the majority of an FG's operational life, it will be serving as a bias in a circuit. In other words, not operating in programming mode. Therefore, a designer must also consider its operation as a current bias, which unfortunately has a temperature dependence. Furthermore, how can this issue be addressed within the context of a large array of FGs that provides diverse current biases? This chapter presents a low overhead temperature compensation scheme for FG-dense designs.

4.1 Floating-Gate Temperature Dependence

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 [5]. Traditional analog pre-processing stages tend to be highly specialized application-specific systems, but developments in reconfigurable field-programmable analog arrays (FPAAs) [4, 6] 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 [4, 6]. 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 [24]. 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 [25, 26]. 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 chapter 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.

4.2 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 and 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. 4.2). 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 charge to the FG. A more in-depth overview of these programming methods are explained in Chapter 5.

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 4.1 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. 4.2 [26]. 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:

$$I_d = I_o \frac{W}{L} e^{-\kappa V_{fg} q/kT} e^{V_s q/kT} e^{V_d/V_A} \tag{4.1}$$

where all voltages are referenced to the well potential and

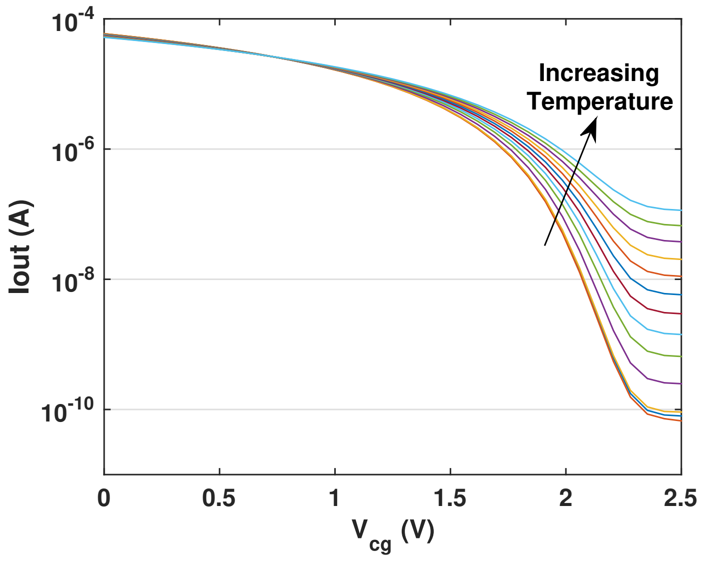


Figure 4.1: Temperature dependence of an FG transistor. The plateaued currents for $V_{cg} > 2.25V$ are artifacts of many junction connections to a single global connection where the current reading was taken. As a consequence, their collective leakage current becomes non-negligible. The current in a single FG transistor continues below these values.

I_0	pre-exponential current scaler;
$\frac{W}{L}$	device dimensions;
V_{fg}	FG voltage;
V_A	Early voltage;
kT/q	thermal voltage;
κ	coupling coefficient from gate to channel that defines the subthreshold current slope;
V_s	source voltage;
V_d	drain voltage.

Equation (4.1) is the same equation as (5.1) with the thermal voltage being expanded here for the context of the temperature effects. Furthermore, V_{fg} as defined in equation (5.2) is repeated here for improved readability:

$$V_{fg} = \frac{Q_{FG}}{C_{tot}} + \frac{C_{cg}}{C_{tot}}V_{cg} + \sum \frac{C_{par}}{C_{tot}}V_x \approx \frac{Q_{FG}}{C_{tot}} + \frac{C_{cg}}{C_{tot}}V_{cg} \tag{4.2}$$

where

Q_{fg}	total charge present on the FG;
C_{tot}	sum total of all capacitances coupled to the FG including parasitics;
C_{cg}	control gate capacitance;
C_{par}	parasitic capacaitences;
V_x	the respective terminal voltages that correspond to C_{par} coupled to V_{fg}.

The right-hand side is a reasonable approximation given that C_{cg} — shown in Fig 4.2 — represents the majority of total capacitance C_T. Then, incorporating (4.1) and (4.2) into the FG current multiplier topology of Fig. 4.2 renders the following output current relationship:

$$I_{M_1} \approx I_{_{REF}} exp \frac{q\kappa(Q_{M_{REF}} - Q_{M_1})}{C_T kT} \tag{4.3}$$

where

$Q_{M_{REF}}$, Q_{M_1}	amount of charge on their respective FGs.

Equation (4.3) 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 equation (4.3) as β with the exception of T

$$\beta = \frac{q\kappa(Q_{M_{REF}} - Q_{M_1})}{kC_T} \tag{4.4}$$

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

$$I_{M1} = \begin{cases} I_{_{REF}} e^{\beta/T} & Q_{M_1} < Q_{M_{REF}} \Rightarrow I_{_{M_1}} > I_{_{REF}} \\ I_{_{REF}} e^{-\beta/T} & Q_{M_1} > Q_{M_{REF}} \Rightarrow I_{_{M_1}} < I_{_{REF}} \end{cases} \tag{4.5}$$

Taking the derivative of (4.5) with respect to temperature yields a negative temperature relationship for $I_{M_1} > I_{_{REF}}$ and a positive relationship for $I_{M_1} < I_{_{REF}}$.

$$\frac{dI_{M1}}{dT} = \begin{cases} -\frac{I_{REF}\beta}{T^2} e^{\beta/T} & Q_{M_1} < Q_{M_{REF}} \\ \frac{I_{REF}\beta}{T^2} e^{-\beta/T} & Q_{M_1} > Q_{M_{REF}} \end{cases} \tag{4.6}$$

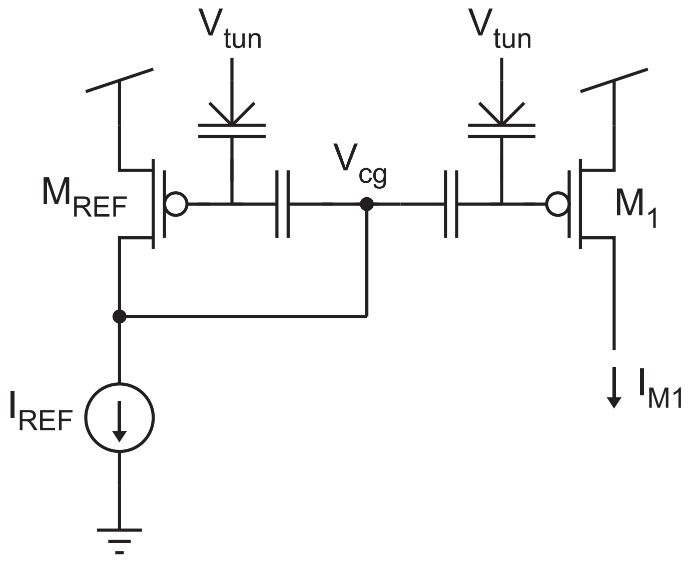


Figure 4.2: Floating-gate current mirror configuration.

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

4.3 FG Temperature Compensation

The application presented in this discussion is applied to our FPAA, which is called the Reconfigurable Analog and Mixed-signal Platform (RAMP) [4]. To provide temperature compensation to such a large-scale system, we are leveraging the FG current mirror shown in Fig. 4.2. 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.

4.3.1 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 4.3(a) 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 4.3(b) 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 (4.2), 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.

4.3.2 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 [27]. 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. 4.4. This shows the temperature compensation performance relative to the room temperature programming target from $-25°C$ to $85°C$. Each line in Fig. 4.4(a) corresponds to a different ratio between the current I_{REF} and the current flowing through transistor M_n (depicted in Fig. 4.3(b)). The current I_{Mn} was programmed at room temperature ($25°C$) and adjusted via the FG temperature compensation structure shown in Fig. 4.3.

The large variance in current ratios shown in Fig. 4.4 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 (4.3), 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 (4.3). However, despite not working as well as a $1:1$ ratio, these cases still perform better than an umcompensated scenario, as shown in Fig. 4.4(c).

4.4 System Performance

Figure 4.1 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}.

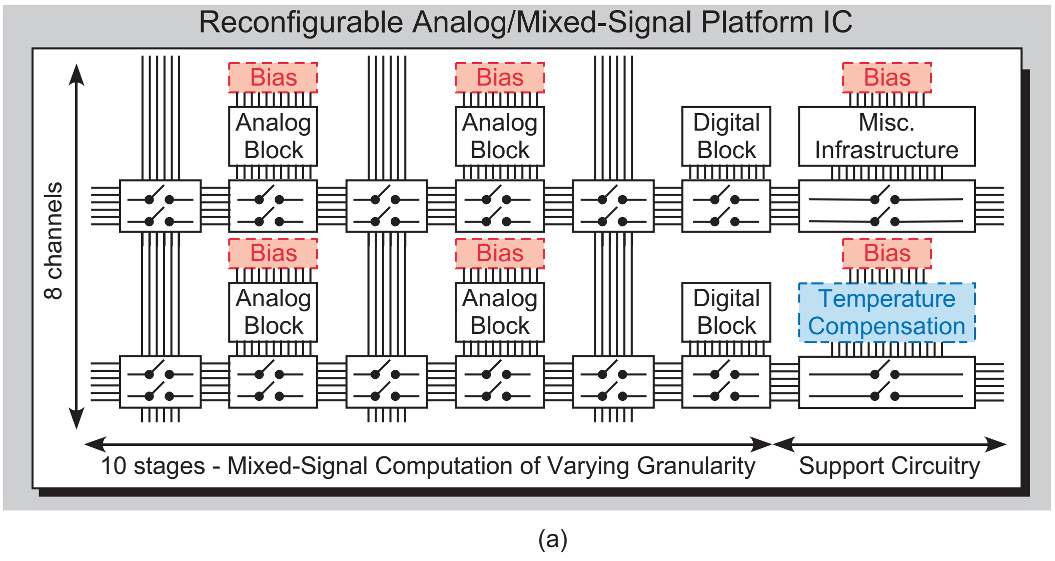
To show the advantage of temperature compensation in the RAMP system, a current-controlled ring oscillator has been synthesized from one of the CABs. This circuit utilizes an input current, I_{in} (generated by one of the FGs) to starve its odd number of inverters, producing output frequency oscillations proportional to the input current (Fig. 4.5(a)). 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 FG. The oscillator output frequency for both cases is shown in Fig. 4.5(a). 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. 4.6. 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. 4.5 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.

4.5 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.


Reconfigurable Analog/Mixed-Signal Platform IC
Bias
Analog Block
Digital Block
Misc. Infrastructure
Temperature Compensation
8 channels
10 stages - Mixed-Signal Computation of Varying Granularity
Support Circuitry
(a)

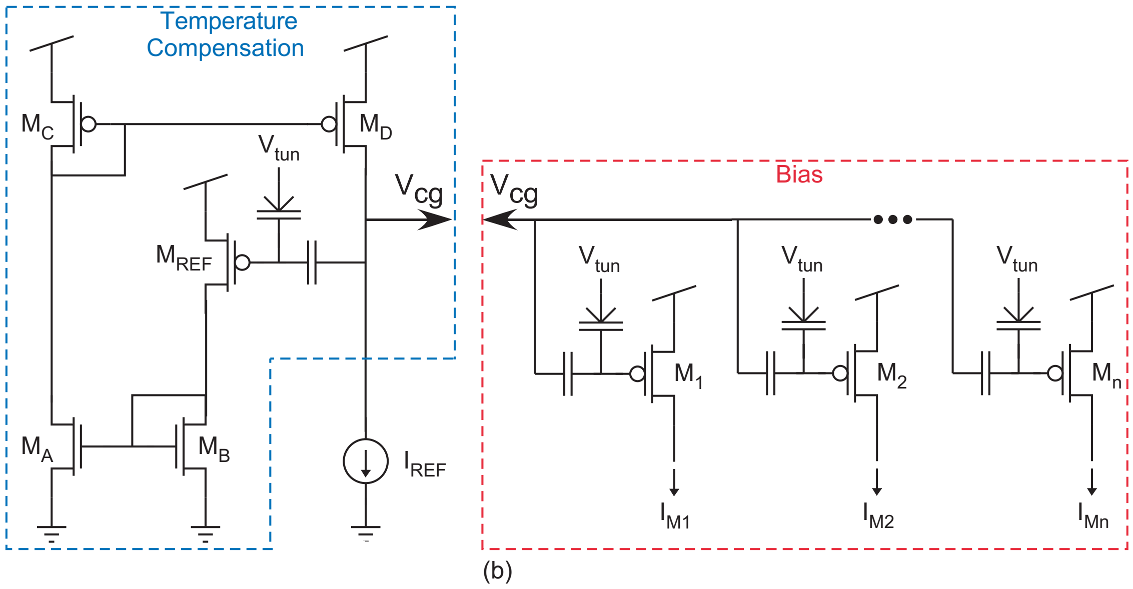


Figure 4.3: (a) FPAA block diagram showing the position of the specialized CAB which houses the FG temperature compensation structure. (b) Floating-gate temperature compensation structure showing connection to global V_{cg}. Currents source to an nFET current mirror before connecting to the CAB circuits.

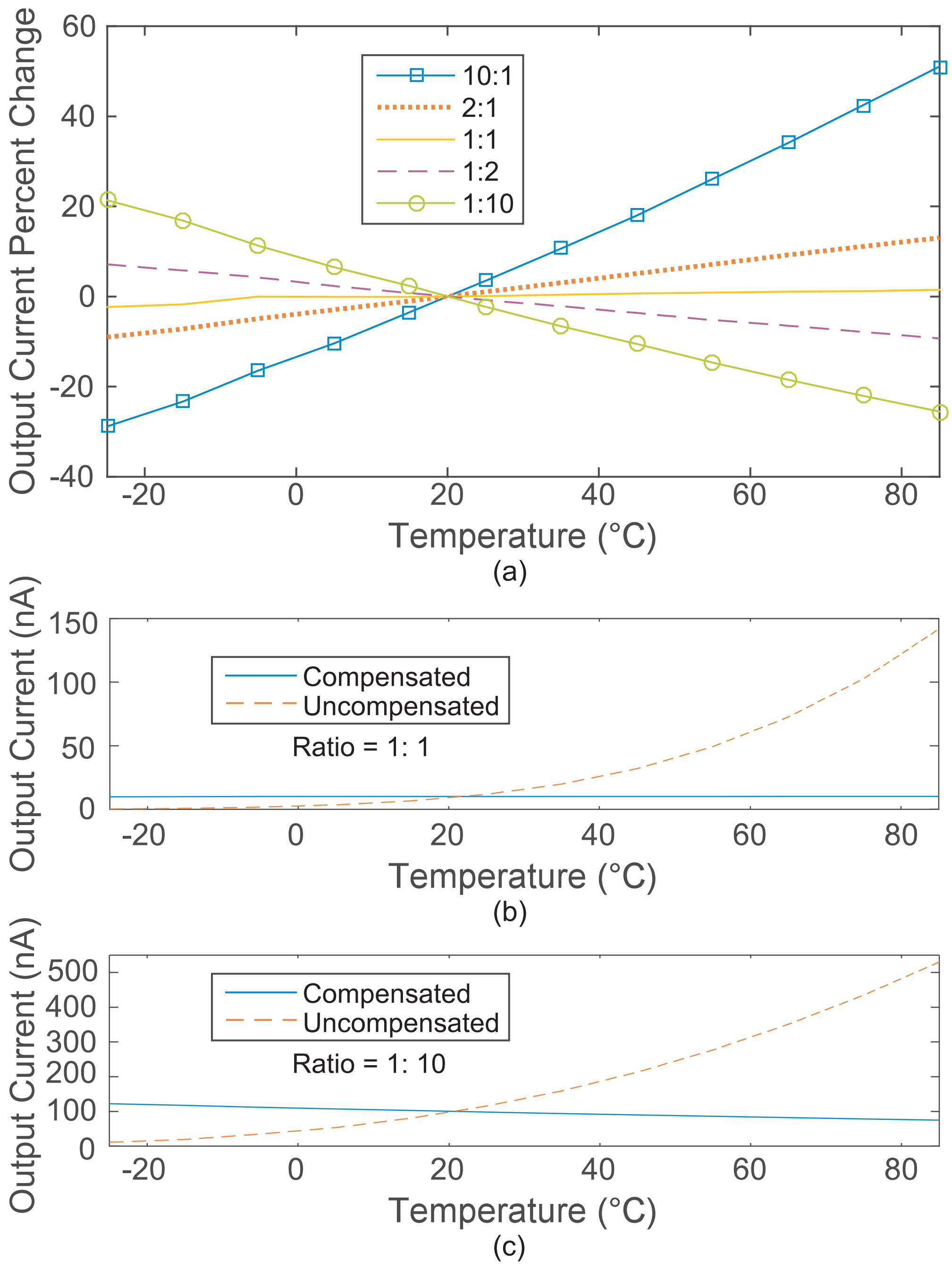


Figure 4.4: (a) Percent change in output current of an FG with temperature compensation normalized to room temperature. (b) and (c) show the effects of compensation compared to an uncompensated case for a ratio of 1:1 and 1:10 respectively.

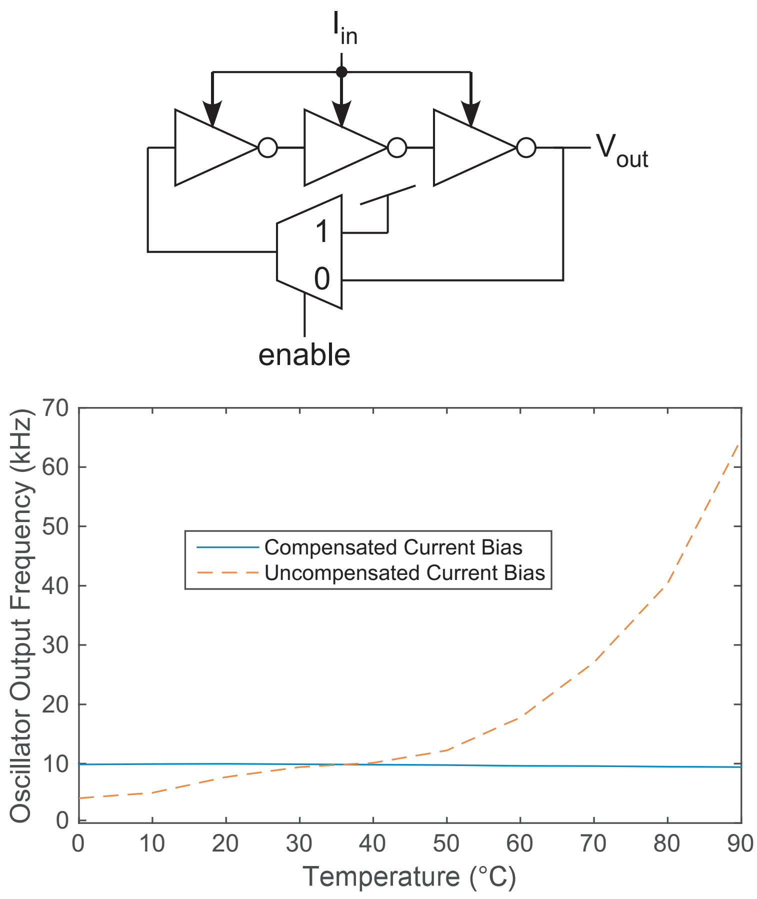


Figure 4.5: Ring oscillator frequency output with respect to temperature for a compensated and uncompensated FG current bias.

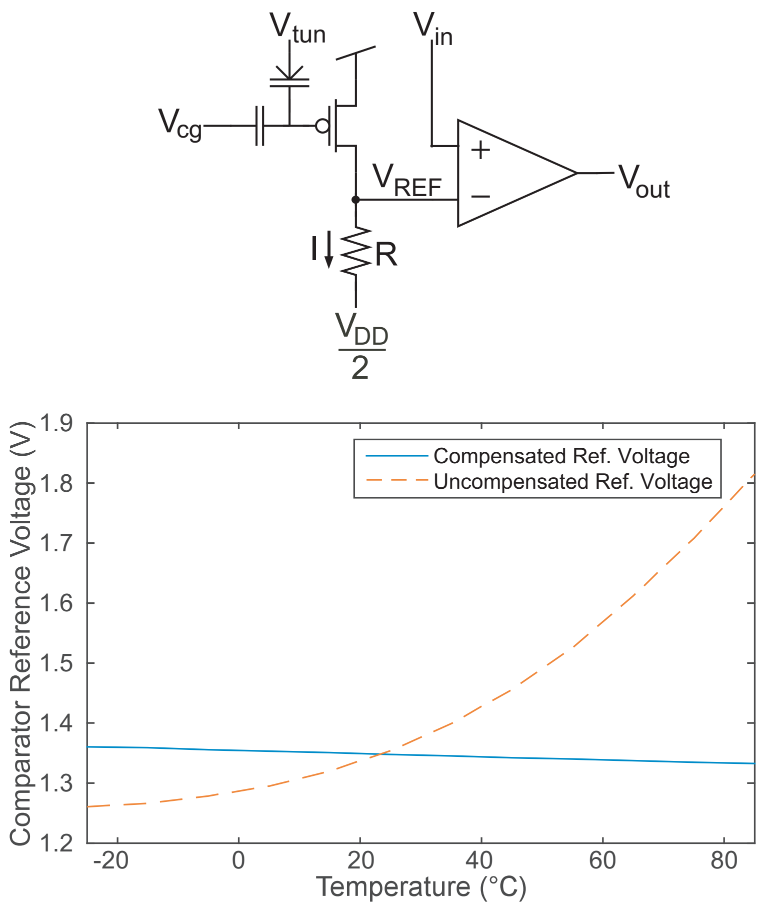


Figure 4.6: Programmed comparator reference value with respect to temperature for a compensated and uncompensated FG current bias.

Chapter 5

Floating-Gate Device Introduction

Before presenting research on FG programmers, we need to establish a requisite knowledge base of the device. This chapter presents the origins of FGs and their applications and development over the years. Furthermore, we'll describe the device's physical make-up and how that affects its operation, and the mechanisms that permit its programmability.

5.1 Floating-Gate Development and Background

Non-volatile memory (NVM) is probably the most familiar application of floating-gate (FG) transistors which was discovered by Kahng and Sze in 1967 [28]. Since then, NVMs found in such products as flash memory and solid-state drives have received a large research focus and have become ubiquitous in consumer electronics in the past ten years. As the name suggests, NVM distinguishes itself from it's "volatile" counterpart (most conventionally used as computer memory) in that it holds its state or charge after it loses power.

FG transistors are charge-holding devices whose gate does not have a DC path to ground (i.e. capacitively held), which makes them great candidates for holding charge for long periods of time. In fact, it has been shown that the change in charge is $< 1\%$ over a 10 year period [29, 30]. Excellent charge retention and the consistent downscaling in device dimensions have made it a pervasive in digital applications [31]. Employing FGs for digital purposes literally means adding/removing enough charge to the FG to measure a '0'/LOW or a '1'/HIGH at the output. This is akin to measuring a digital output voltage where a '0' is ground and a '1' is circuit's supply voltage. However, much in the same way there is a spectrum of analog voltage outputs between a '1' and a '0', FGs can be employed for analog applications since there is a continuum of charge that can be placed on the FG.

The amount of charge placed on a FG can be quantitatively modeled as a smooth continuous function allowing it to be used as a computational element since this charge amount can be controlled by a programming process. Much in the same way of digital flash applications benefited from FG dimension downsizing, analog FGs stand to benefit as well [32]. Some of the first applications of analog NVM were largely biologically-inspired. In 1989, Mead presented a retina circuit and cochlear model [33] and Intel offered their electrically trainable artificial neural network chip (ETANN) that utilized an array of FGs to create learning synapses [34]. In 1992, a 'nueron MOS' was presented because of its behavior that was markedly similar to neurons [35].

Many of these biologically-inspired circuits laid the groundwork for the synthesis of FGs into traditional analog circuits: translinear and log-domain computations [36, 37, 38], adaptive voltage taps for analog-to-digital converters [39], and input-offset compensation for amplifiers [40]. FGs have also offered compelling solutions for post-fabrication mismatch through trimmable current sources [41, 42, 43] and pin-count reduction with on-chip biasing [37].

Many of the aforementioned FG usages have been application-specific or supplemental to a circuit's function with the exception of the ETANN chip. The ETANN chip has become the archetype of contemporary system-level FG design for its arrays of FGs used to create individual computational blocks. However, the impetus for modern system-level designs have been to gain the benefits of reconfigurability for analog design and analog signal processing. As a result, the modern realizations follow in spirit of Field-Porgrammable Gate Arrays (FPGAs), and these analog counterparts are referred to as Field-Programmable Analog Arrays (FPAAs) [44]. Over time though, FPAA platforms have since been developed to harmonize the analog/digital divide onto a single chip to complement the circuit needs for the application space [4],[7], [45].

Given all these highly sought after analog applications, one would conceive FGs are highly emphasized and vastly deployed in analog design. However, analog FGs have been slow to be integrated into designs due to practicality of programming. Many of the aforementioned research designs need to be programmed with fine granularity which involves intimate physical device knowledge, expensive laboratory equipment (many high-precision voltage/current references), and extensive support circuitry. Because of this high barrier to entry, improving the support circuitry became the inspiration to a lot of the designs in this work.

5.2 Floating-Gate Device Overview

A FG transistor in this work refers to a p-channel MOSFET (pFET), whose gate is isolated via opposing oxides and does not have any direct connections to a voltage potential. A schematic of a FG transistor juxtaposed with a conventional pFET is displayed in Fig. 5.1 along with the physical representation. From the physical cross-section, we can see that the FG is essentially has an additional gate-oxide stacked on top of a conventional pFET to render the FG. Interestingly, the floating gate utilizes polysilicon material (i.e. not a metal material) which is better for long-term charge storage as it will not leak charge over time.

A few other distinctions from a conventional pFET is that the two gates on the FG must be distinguished from each other. The node that is accessible to the designer is referred to as the control gate V_{cg}. The control gate is the gate that will be used in the same manner as V_g of the pFET to operate the FG device. Additionally, there is a parallel plate structure in the FG formed by the gate-oxide-gate stack. Because of this structure, it is schematically modeled as capacitor C_g. And finally, V_{fg} refers to the effective floating gate voltage which will be the node that stores the charge.

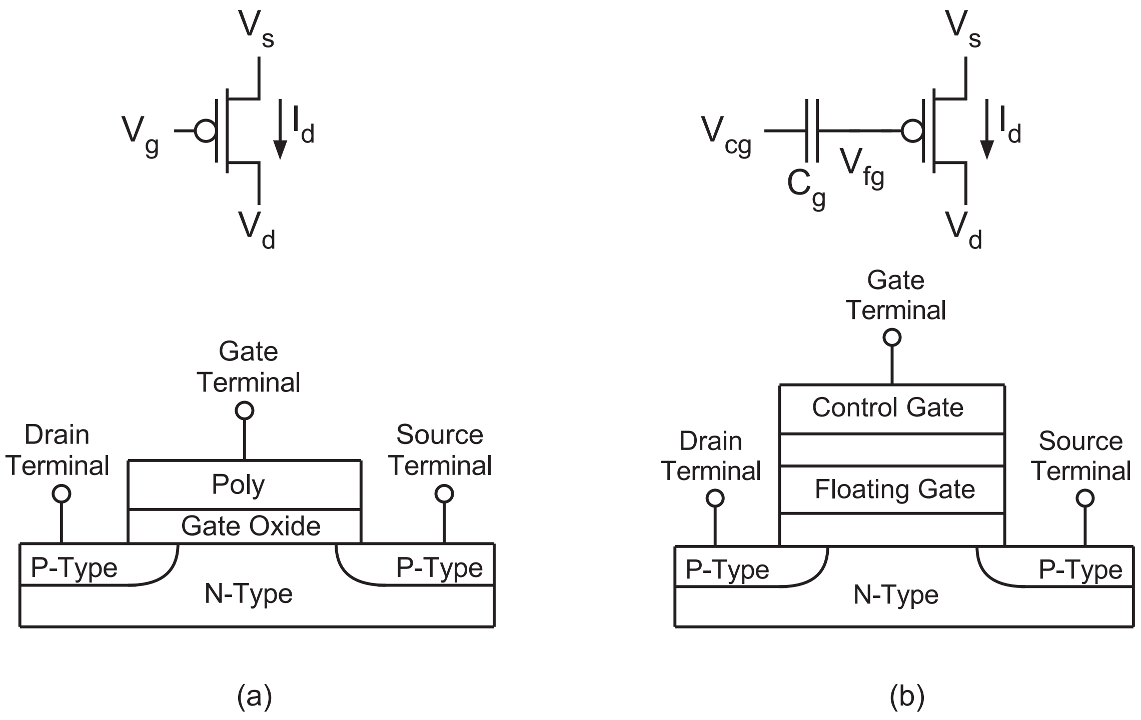


Figure 5.1: Schematic representation and physical cross-section of (a) a pFET transistor and (b) a FG transistor, respectively.

For a FG to be used computationally, the amount of charge on the FG must be modifiable. The process of modifying charge on a FG is referred to as FG programming and this distinction is made because of the unusual nature of a capacitively isolated node. In this work, 'FG programming' refers to electrically programming a FG to add and remove charge. The main approach to create electrical programming conditions is to distort the FG energy band diagram such that there is a great probability for charge transport to/from the floating gate. For most designers to accomplish the energy band distortion, they employ one of two quantum mechanical processes: Fowler-Nordheim (FN) tunneling or hot-electron injection (HEI).

In the earlier years of FG developments, charge was added to floating gates via FN tunneling. If the devices were not of the type one-time programmable (PROM), they could be erased and re-programmed again (EPROM). EPROM devices were globally erased by exposure to UV light which exploited the photoelectric effect to remove the electrons. After erasure, EPROM devices could be reprogrammed, and by this point in time, many devices were utilizing HEI as their method of choice for adding electrons to the FG. Further developments allowed for electrically programmable and erasable devices (EEPROM), which obviated the need for UV erasure. EEPROM devices have many FGs arrayed and addressable in a NOR flash configuration for the designer. This configuration permitted HEI to add charge to the FG selectively (i.e. program a specific FG) and globally erase all devices via FN tunneling. This is exactly the method that is used to add/remove charge from the FG devices in this work.

The effects of HEI and FN tunneling are visually illustrated in the analog FG context in Fig. 5.2 in terms of an IV-curve. For conceptual example, the effect of electron removal via tunneling is represented in an IV-curve shift from the pre-tunneled green curve to the post-tunneled blue curve. In essence, to achieve the same output current after tunneling as before tunneling — for example at 1μA — the V_{cg} must be lowered to compensate for the loss of negative charge. The opposite is true for the HEI process. Additional negative charge on V_{fg} effectively pulls V_{fg} lower, resulting in larger V_{cg} voltages needed to achieve its pre-injected current position. More generally, the shifts in the IV curve can be viewed as a change in the threshold voltage of the FG from the perspective of the control gate.

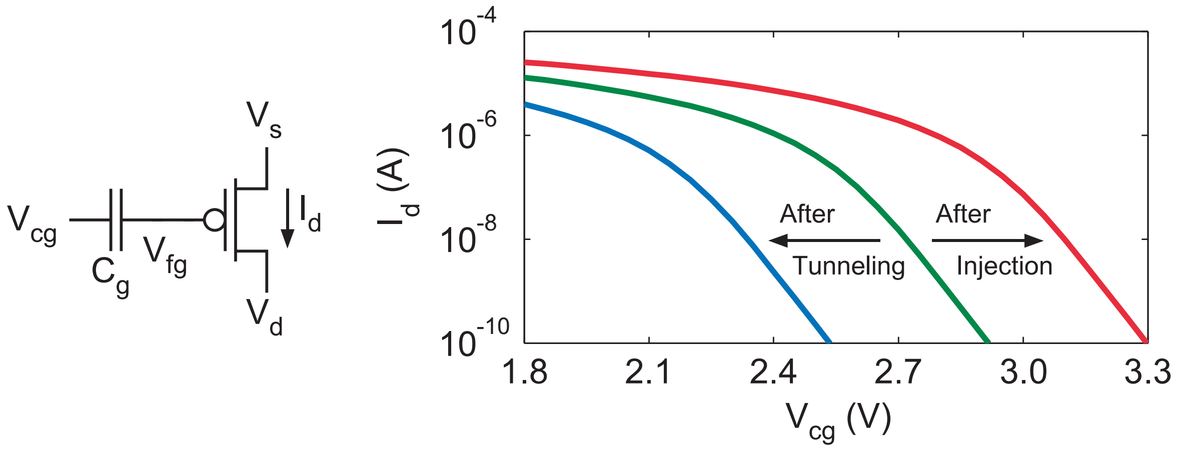


Figure 5.2: Floating-gate transistors allow for programmable threshold shifts from the perspective of the control gate. Injection programming adds electrons to the floating-gate, while tunneling removes electrons from the floating-gate.

5.3 Floating-Gate Device Structure and Operation

The schematic and cross-section of a FG Fig. 5.1b is a generic representation of the device does not fully encompass the devices used in this work and their connection to programming. A more accurate schematic model is shown in Fig. Fig. 5.3. In fact, the FG devices deployed on all designs described in this work include a supplementary node called the tunneling node V_{tun}. From the schematic of the device, the V_{tun} is a second interface to the floating gate node V_{fg} via a varactor that will be activated during FN tunneling. The varactor is realized with a pFET MOSCAP (i.e. shorted source, drain, and well connection) which can be seen in the physical cross-section. A pFET MOSCAP was chosen over other capacitors, including nFET MOSCAPs, for its more efficient charge transport, lower power consumption, and consistency across various CMOS processes [46]. The pFET MOSCAP still conforms with the aforementioned specification that the floating gate must be implemented in a non-metal material like polysilicon. Furthermore, from Fig. 5.3 it's shown that the control-gate capacitor plates are of polysilicon-insulator-polysilicon type visually represented as light grey and dark blue. The last component of the FG schematic is the pFET transistor; it is represented as the left device in the physical cross-section. It has its own n-well, but shares the floating gate poly1 connection with the MOSCAP.

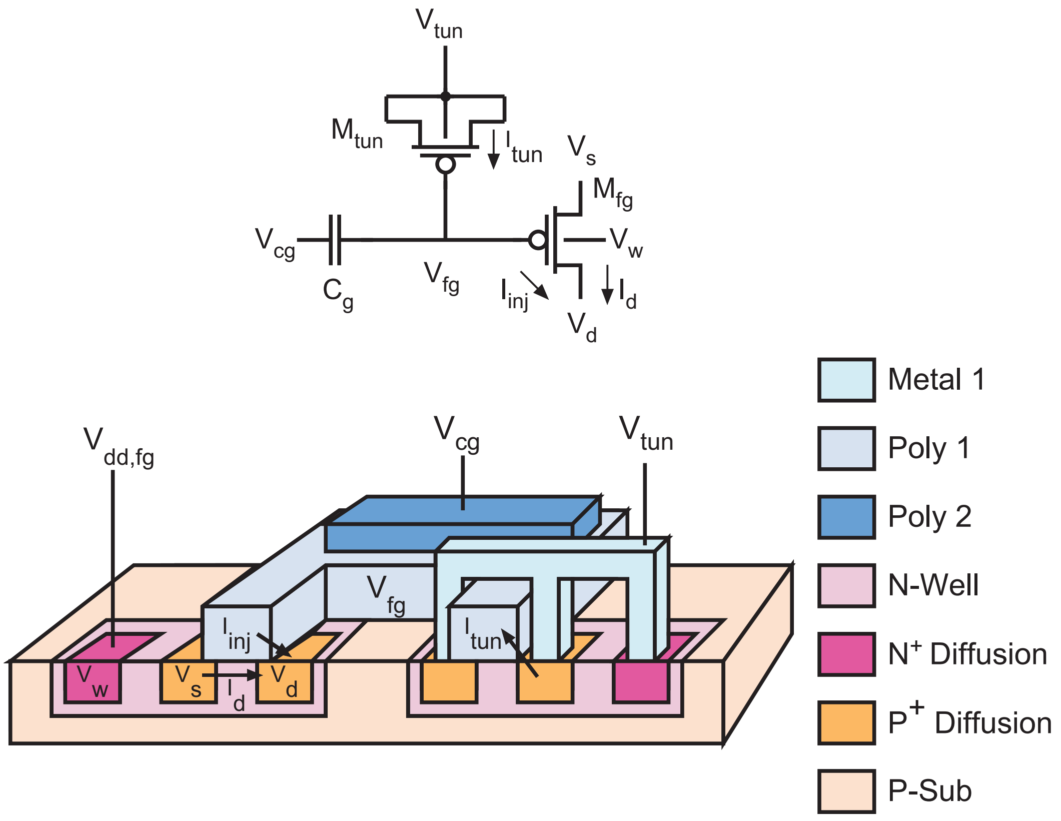


Figure 5.3: Physical cross-sectional layout of a FG transistor.

5.3.1 Floating-Gate Operation

FGs utilized in this work are operated in the subthreshold region — a weak inversion region — where $V_{sg} < |V_{thp}|$. While introductory texts of MOSFETs largely describe $V_{sg} < |V_{thp}|$ from an operational standpoint as 'OFF', when in actuality a V_{sg} voltage very near but below $|V_{thp}|$ is an edge-case where there is still current conduction. The rationale being that the current is 'OFF' in that the channel current measured in weak inversion is magnitudes lower relative to strong/moderate inversion. Subthreshold current was initially been viewed by analog designers as a parasitic leakage current since the device was not being used but drew currents. However, as demand for lower-power design has significantly increased, many designers have leveraged this region of operation for supremely large power savings in their designs.

The drain current of a pFET in the subthreshold saturation is expressed as follows:

$$I_d = I_0 \frac{W}{L} \exp\left(-\frac{\kappa V_g}{U_T}\right) \exp\left(\frac{V_s}{U_T}\right) \exp\left(\frac{V_d}{V_A}\right) \tag{5.1}$$

with all voltages are referenced to the well voltage, where

I_0	pre-exponential current scaler;
$\frac{W}{L}$	FET dimensions;
κ	coupling coefficient from gate to channel;
V_g	gate voltage;
U_T	thermal voltage $\approx 26mV$ at room temperature;
V_s	source voltage;
V_d	drain voltage;
V_A	Early voltage.

Like the above threshold case, we can see in (5.1) that there is a relation between V_{sg} and the channel current which has made subthreshold operation viable. The relationship in (5.1) is not governed by square law, but is exponential since the dominant current component is diffusion current [47]. This means that a small movement in V_{sg} away from V_{th} results in a precipitous drop in output current. This is an aspect of subthreshold conduction that it shares with BJT conduction. In fact, subthreshold conduction boasts the highest ratio of transconductance-to-channel current out of all regions of operation in MOSFETs. This is only slightly lower than BJT's transconductance-to-channel current ratio [47].

Next, we need to incorporate the FG element into (5.1). The first modification is V_g becomes V_{fg} for the FG. The FG's operation will be dictated by the voltage V_{fg} which is made up by the charge on the FG divided by the total effective capacitance plus/minus the voltage coupled from the V_{cg}. The total capacitance C_{tot} is the sum of all capacitances interfaced with the FG including all parasitic capacitances. All the capacitors referred to the floating-gate are visually represented in Fig. 5.4 showing the tunneling and control gate capacitor in addition to the physical parasitic capacitances. Modeling V_{fg} can be expressed in terms of a capacitive divider where each corresponding capacitance/voltage pair contributes to the overall V_{fg}. Each capacitive/voltage pair is more generally referred to as C_x and V_x in the following expression:

$$V_{fg} = \frac{Q_{fg}}{C_{tot}} + \sum \frac{C_x V_x}{C_{tot}} \tag{5.2}$$

$$V_{fg} = \frac{Q_{fg} + C_{cg}V_{cg} + C_{tun}V_{tun} + C_s V_s + C_d V_d + C_{well}V_{well}}{C_{tot}} \tag{5.3}$$

where

Q_{fg}	total charge present on the FG;
C_{cg},C_{tun}	node capacitances (see Fig. 5.4);
C_s,C_d,C_{well}	parasitic capacitances (see Fig. 5.4);
C_{tot}	sum total of all capacitances coupled to the FG.

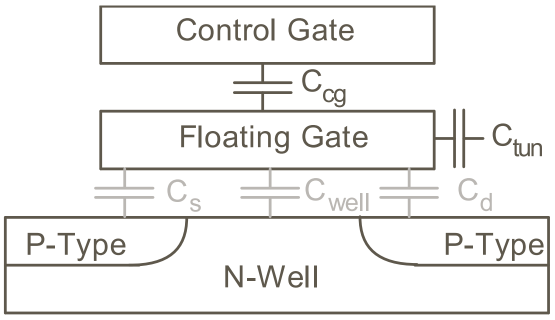


Figure 5.4: Physical cross-sectional layout of a FG transistor showing all capacitors coupled to the FG including parasitics (in gray color).

However, when taking into consideration each capacitor's contribution to the overall C_{tot}, over 90% of the capacitance is dominated by C_{cg} at $100f$F. The next largest contribution is C_{tun}, but V_{tun} is always LOW (unless there is tunneling), and the parasitics are magnitudes lower in capacitance. (5.3) can be accurately approximated to the following expression:

$$V_{fg} \approx \frac{Q_{fg} + C_{cg}V_{cg}}{C_{tot}} \tag{5.4}$$

We can now substitute (5.4) into (5.1) to completely model a FG for subthreshold which renders the following expression:

$$I_d = I_0 \frac{W}{L} \exp\left(-\frac{\kappa}{U_T}\frac{Q_{fg} + C_{cg}V_{cg}}{C_{tot}}\right) \exp\left(\frac{V_s}{U_T}\right) \exp\left(\frac{V_d}{V_A}\right) \tag{5.5}$$

From (5.5), there is still have a controllable voltage node V_{cg} to modulate the drain current like V_g in (5.1). What's been gained with the FG functionality is the Q_{fg} factor gives a designer the ability to choose a threshold voltage after fabrication with FG programming. The programmable threshold shift is demonstrated in Fig. 5.2.

5.4 FG charge modification

5.4.1 FG Tunneling and Electron Removal

Referring back to Fig. 5.3, the tunneling current labeled on the MOSCAP indicates that FN tunneling process removes electrons from the FG. In order for electrons to be transported across the oxide barrier, the band diagram needs to distorted at the oxide interface (i.e. by creating an electric field across it) to effectively reduce the barrier width. This charge transport mechanism is a probabilistic event, and therefore, the smaller the barrier width, the higher the probability for the electron to cross the barrier. A figure demonstrating FN tunneling through the pFET MOSCAP on a FG device is shown in Fig. 5.5. Employing a pFET MOSCAP is the typical method of realizing a tunneling connection. The FG schematic representation has the same orientation to match the corresponding energy band diagram. In this instance, a large electric field bends the oxide barrier at V_{tun} which has the source, drain and body shorted to a single node. The effective width of the barrier needs to be reduced to at least $5nm$ for the FN tunneling process to have a measurable effect [30].

The tunneling current can be expressed by the follow equation [46]:

$$I_{tun} = \alpha exp \frac{-\beta t_{ox}}{V_{ox}} \tag{5.6}$$

where t_{ox} is the oxide thickness, V_{ox} is the potential difference between V_{tun} node and V_{fg}, and α and β are constants that depend on the CMOS process and the type of device one is tunneling through. To prevent undesirable tunneling in smaller CMOS processes, where the standard device oxide thickness is smaller than $5nm$, typically FGs are implemented with I/O high voltage devices [48].

The top plot of Fig 5.6 shows the effective tunneling voltage for standard CMOS processes. Included in the same plot shows the scaling of V_{dd} over the various CMOS processes. The plateauing of V_{tun} at 250nm and below can be contributed to FG devices being realized with I/O devices as opposed to their standard CMOS device. Much like the scaling of feature sizes, the oxide thicknesses of the respective processes have downsized too. The standard FETs of the smaller technologies have thin oxides that are not conducive to retaining charge for long periods of time. In the bottom plot of Fig. 5.6, we see the ratio of V_{tun}/V_{dd}. For larger devices (i.e. $< 250n$m), the ratio stays relative constant, while smaller technologies have seen an expected rise in ratio due to the usage of I/O devices. The dashed ratio line in the bottom plot shows that a lower supply V_{dd} should be employed for FGs as opposed

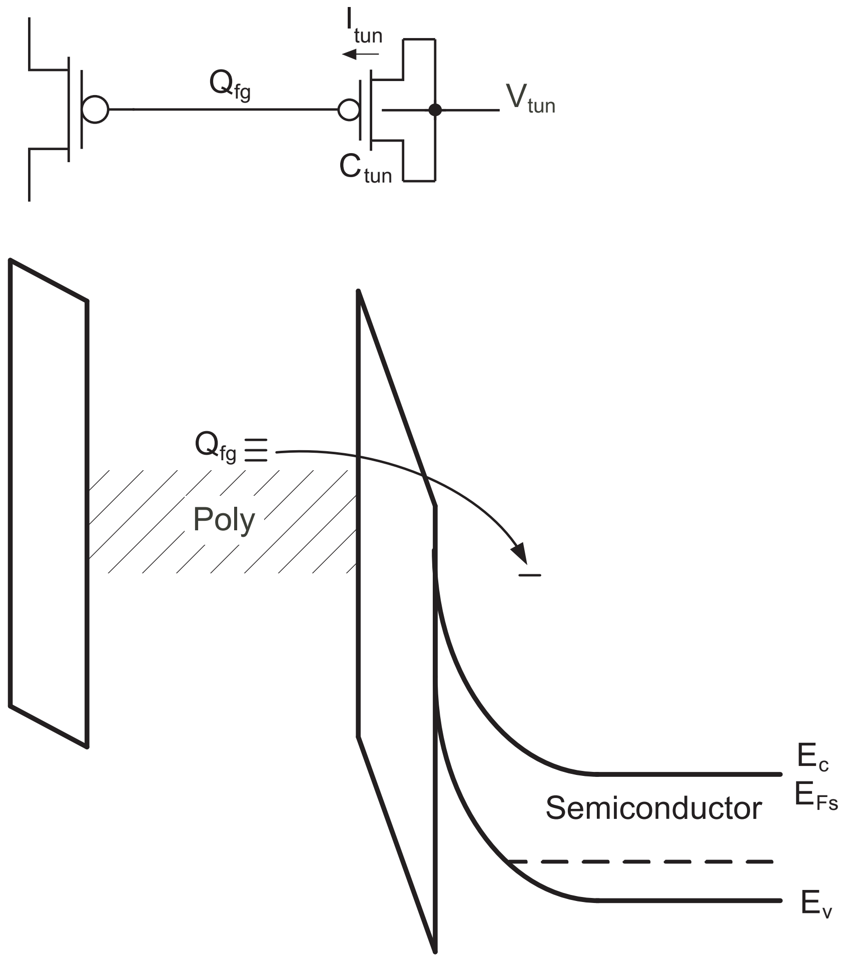


Figure 5.5: Qualitative demonstration of Fowler-Nordheim tunneling in the FG band diagram. The control gate has been omitted in the schematic for clarity.

to the stated nominal V_{dd} of top plot. This is due to unwanted HEI that can occur at the standard supply V_{dd} which can add additional charge to the FG. Hence, the practical ratio line is given, but is not relevant to smaller technologies using I/O devices.

In conclusion, tunneling for this work is used to remove electrons from a FG. The net result of having less negative charge on the FG effectively reduces the threshold voltage from the perspective of the control gate V_{cg} and confirms the movement of the blue IV-curve to the left in Fig. 5.2. Tunneling is typically used as a global charge erasure since circuit configurations are limited and make it difficult to isolate to just a single FG among many on the same chip.

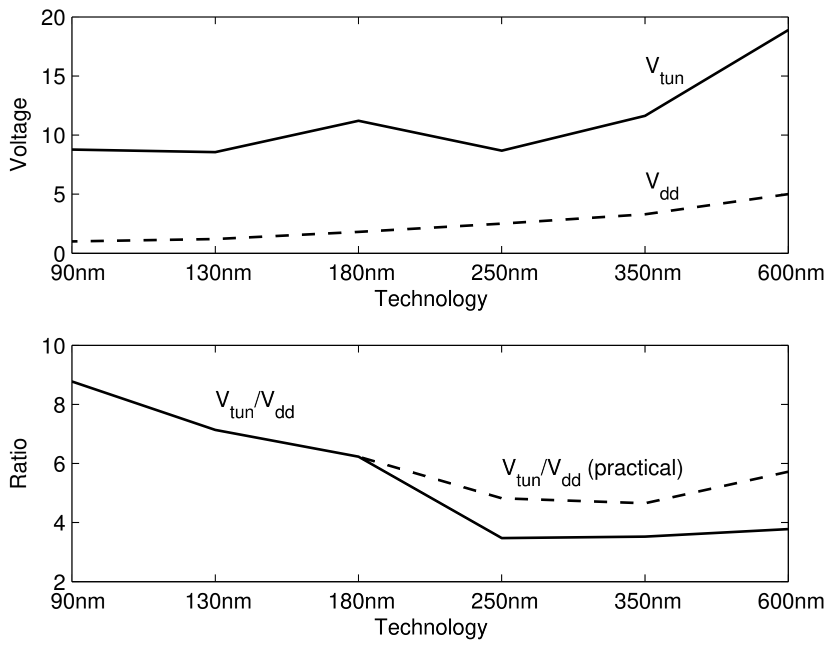


Figure 5.6: Comparison of the required V_{tun} voltage for scaling CMOS technologies. Top plot shows that the V_{dd} has scaled for the technologies, but V_{tun} plateaus at 250nm due to FGs needing to be realized in less leaky I/O devices. Bottom plot shows the ratio of V_{tun} to V_{dd}. Above 250nm, the ratio is relatively constant. 250nm and below, the ratio increases because V_{dd} continues to scale as V_{tun} stays relatively constant [1].

5.4.2 Hot-Electron Injection and the Addition of Electrons

The complementary process of FN tunneling is to add electrons to the FG by hot-electron injection (HEI). The term 'injection' refers to an electron having enough energy to overcome oxide barrier and inject onto the FG from the drain/channel interface. While somewhat similar to FN tunneling in that an electron tunnels through the oxide barrier, the event is created under a different set of circumstances. Additionally, it must be reconciled that the context of this discussion is charge modification of pFET FGs, whose majority carriers are holes. So, in order for an electron to surmount the oxide barrier, the electron must first be sourced by an electron-hole pair generation. The conditions suitable for HEI stem from a series of probabilistic events.

The main impetus for creating highly energized electrons to overcome the oxide barrier is a large potential difference from the source to drain nodes. This will in turn create an strong electric field, with especially strong electric field lines at the interface of the drain depletion region interface with the channel (i.e. pinch-off region). It is at the drain interface that the conducting holes will become highly energized and some will collide with the lattice. It is from these collisions that a subset of them will have enough force to promote a valence band electron to the conduction band, and consequently generate a new electron-hole pair. Even at this point in the process, it is not guaranteed that the newly freed electron will inject on to the FG. It is only if the resulting 'hot' electron has enough energy to surmount the oxide barrier. This process is illustrated in Fig. 5.7. While the diagram is not to scale, the depletion region between the drain and channel demonstrates how impact ionization is possible given the amount of band distortion.

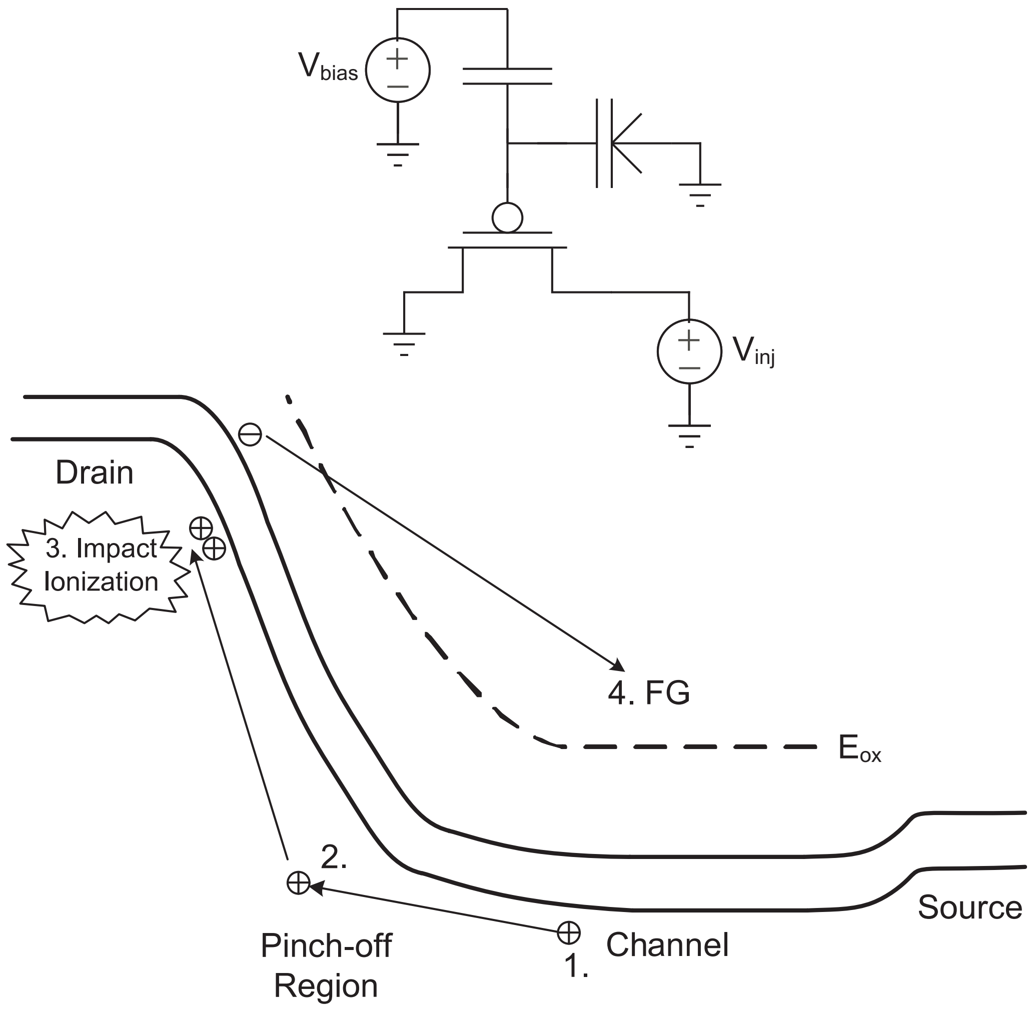


Figure 5.7: Demonstration of hot electron injection in the FG band diagram. The band diagram is not to scale.

The HEI current has been characterized to be

$$I_{inj} \approx \beta I_s \exp\left(\frac{V_{cd}}{V_{inj}}\right) \tag{5.7}$$

where I_s is the source current in the subthreshold range, V_{cd} is the channel-to-drain potential, and β and V_{inj} are fit constants [49]. From this equation, there are two factors that the designer has control over: I_s and V_{cd}. Referring back to to 5.7, one can apply a voltage at V_{cg} to set a subthreshold current (i.e. typically sub-1μA). Focusing on the latter factor, there is an exponential relation for the HEI current to V_{cd}, which in practice is set by one's applied V_{sd}. This applied V_{sd} is outside the normal supply rails between the source and drain for favorable HEI conditions. This why operating the FG under normal conditions contributes negligible HEI current. The graphs in Fig. 5.8 show the varying V_{sd} voltages needed injection over different technologies.

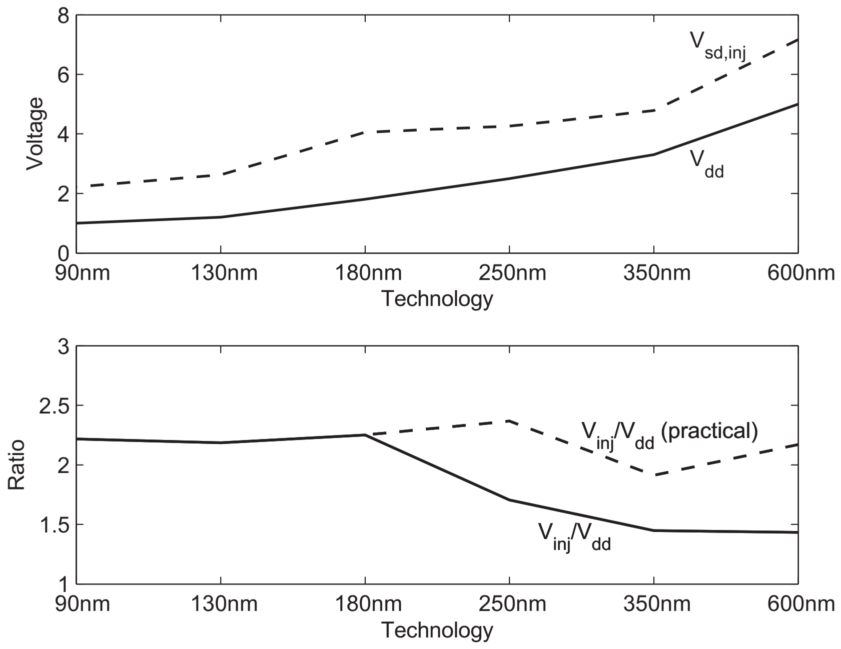


Figure 5.8: Comparison of the required V_{inj} voltage for scaling CMOS technologies. Top plot shows that V_{dd} and V_{inj} scale similarly over the downsizing of technologies. Bottom plot shows the ratio of V_{inj} to V_{dd}. In practice, the supply V_{dd} should be kept lower than standard V_{dd} to prevent unwanted HEI. Taking this into consideration,the ratio stays around 2 over all technologies [2]

.

In the top plot of Fig. 5.8, V_{inj} and V_{dd} follow a similar slope throughout the scaling. This behavior is somewhat different from V_{tun} characterization of Fig. 5.6. In Fig. 5.8, the scaling does not plateau for smaller technologies that use thicker oxide I/O devices. We can contribute this to injection's larger dependency upon the source/drain junction depth, which scales smaller as the technology shrinks [50]. When examining the ratio of V_{inj} to V_{dd} in the bottom plot shows a relatively constant for smaller technologies who employ I/O devices. The larger technologies have a smaller ratio with larger supply V_{dd}s. In practice and similar to Fig. 5.6, the operational V_{dd} should be lowered for the larger technologies due to the possibilities of unwanted HEI at the standard supply V_{dd}s.

Overall, HEI has the net effect of adding electrons to the FG and increasing the threshold voltage from the perspective of the control-gate. This makes sense since this process adds more negative charge to the FG. As a result, this will require a larger V_{cg} to offset said charge to remove the conducting inversion layer if you wanted to turn the device "OFF". Injection and the increase of threshold voltage is visually represented in Fig. 5.2 by the red line that has shifted to the right from the green line.

Of the two processes, FN tunneling is preferred for erasure since its difficult to isolate specific transistors in an array of FGs. Hot-electron injection is the preferred method to add charge to a FG because for granular charge modification and the ability to isolate the specific FGs. Both HEI and FN Tunneling fall under the category of programming operation. The majority of the time FGs are not being programmed, and they are used under the run regime where they are used for any of the various applications. This is why it's viewed that programming a FGMOS does not structurally compromise the device within the first hundred of thousands of programs. Programming occurs during a short period of time — typically in the range of 10's of milliseconds. In run mode, the FGs utilize the nominal voltage rail as defined by the CMOS technology. However, under the programming regime, the large electric field required to perform injection or tunneling is requires a voltage 2-5 times larger than the nominal voltage supply. As a result, there are a number of peripheral support circuits to assist in programming. This work hopes to serve in improving FG support circuitry.

5.5 Conclusion

FGs are ubiquitous in contemporary computing systems, but are most well-known in the digital domain. However, their nascent analog counterparts can be used for computing as well since they can be programmed to a continuum of values between a logic '0' and '1'. Analog FG design has seen a resurgence of interest as there have been developments in large-scale re-configurable analog systems like field-programmable analog-arrays.

FG programming is typically done in one of two ways: hot-electron injection and Fowler-

Nordheim tunneling. For the scope of this work, HEI is used to add electrons to the FG, and FN tunneling is used to remove electrons from the FG. From an analog designer's perspective, the programming functions as a process to modify a specific device's threshold voltage. In turn, these programmed devices can be used as programmable current sources in circuits.

Chapter 6

Floating-Gate Injection Programming

With the previous chapter detailing the foundations of hot-electron injection, this chapter is devoted to the circuitry that carries out this phenomenon. One of the largest challenges is integrating this process in large-scale FG-dense platforms that achieve programming consistency and accuracy. Furthermore, these goals should be accomplished with reduced requirements — whether that be circuitry overhead or easing the specifications on interfacing circuits — and the ability to perform programming in the field. Towards this goal, we present traditional, above-ground programming methodology before presenting a novel below-ground programming methodology.

6.1 Floating-Gate Injection

Floating-gate transistors are form of non-volatile memory that can be programmed and employed for analog computation (e.g. setting corner frequencies in a band-pass filter). Hot-Electron Injection (HEI) is one of the primary programming methods for adding charge to a floating-gate device. HEI occurs when there holes-impact ionization of a pFET drain, and consequently, the newly generated electron has a chance of being injected on the floating-gate node. A detailed overview of the device physics of HEI is described in Chapter 5.4.2. While the aforementioned chapter described the phenomenon and its conditions, it is not clear how this process is realized in circuitry. This chapter will cover the programmer circuit, which is responsible for carrying out HEI on an FG.

A programmer is a facilitator circuit responsible for carrying out HEI of FGs. While it's possible to program an FG off-chip, it usually provided as an infrastructural element on-

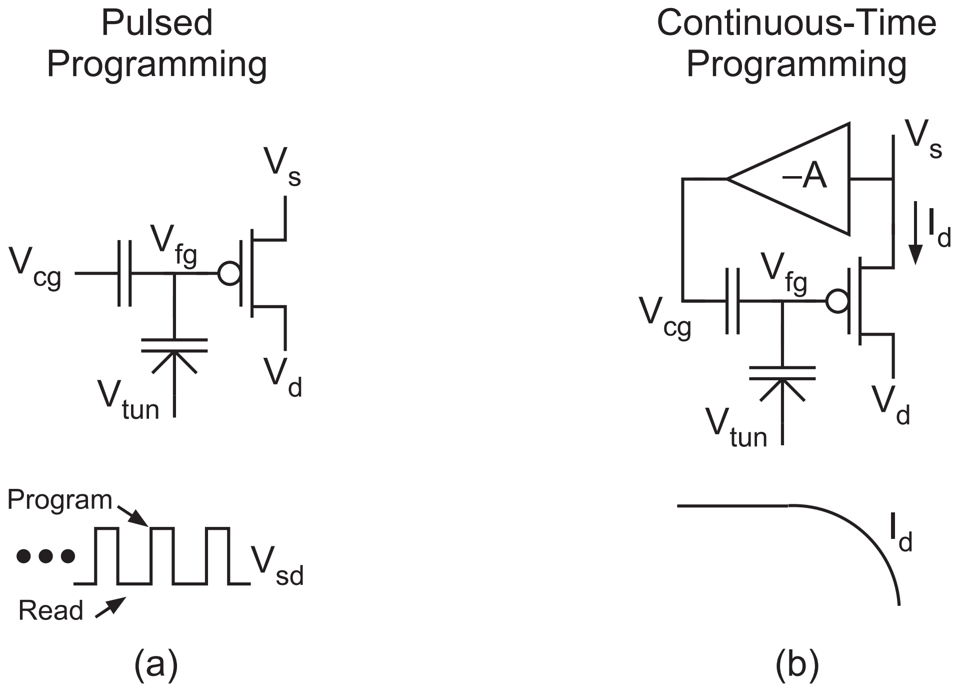


Figure 6.1: (a) Pulsed programming sequentially cycles program/read modes until desired programming target is achieved. (b) Continuous programming employs negative feedback to converge upon programming target.

chip. Generally, designers strive for a programmer circuit to be accurate, have low-overhead, low-power, and the ability to program quickly, and linearly. The most ideal situation is to integrate a programmer on-chip especially when there are many FGs on a single chip like an FPAA that could be used in-the-field [4, 6]. Recalling from equation (5.7), HEI has been characterized and is controllable from the designers perspective with V_{sd}, I_s, and V_{cg}:

$$I_{inj} \approx \beta I_s \exp\left(\frac{V_{sd}}{V_{inj}}\right) \tag{6.1}$$

Programmer circuits run the gamut in techniques for achieving HEI, but can largely be categorized in two distinct types: pulsed-mode programming or continuous-time programming. The two structures are juxtaposed in Fig. 6.1. This chapter will review both regimes, and additionally, various techniques of both including below-ground programming and indirect programming. It will also serve as a comparative study of programmers used for FG-dense structures like FPAAs.

6.2 Pulsed Programming

Pulsed programming is the most accessible methodology of programming an FG since it can be implemented as a simple algorithm. As shown in 6.1(a), there are two modes of operation: read and program. For read mode, the FG is experiencing normal conditions like it would be operated in a circuit application (i.e. a current source). Its purpose is to measure the effect of the preceding injection pulse that was applied during program mode. Program mode creates the conditions favorable for injection by applying a large (i.e. above the nominal supply voltage) V_{sd} pulse. The pulse duration and V_{sd} magnitude are the means for controlling the rate of injection. After a program pulse, the FG is placed in read mode to measure the post-injection current. The process repeats if the desired current is not achieved for a given applied V_{cg} in read mode.

Pulsed programming is largely a brute force methodology for programming FGs. Furthermore, the total duration of program/read cycles required to program an FG will be a function of the programming target, pulse duration and V_{sd} magnitude. In [51], the number of programming cycles needed to achieve the desired programming target is on the order of single thousands. If very high programming accuracy is a design constraint, then clearly this will result in more finer pulses, and consequently a longer programming duration. This overhead compounds when a design contains many FGs that need programmed many times like the training of a neural network circuit synthesized on an FPAA.

There has been progress in pulsed programming to limit the total number of pulses required to program an FG. In fact, [52, 53] can effectively program an FG within eight $20\mu s$ pulses for an optimal V_{sd}. However, to know the optimal V_{sd} requires *a priori* knowledge of each individual FG on the chip. This is done through FG characterization to extract FG parameters which can then be mathematically modeled to yield the optimal V_{sd}. The required characterization adds an initial overhead for on FG-dense chips like FPAA and will be required of every respective chip as there is mismatch among chip-to-chip and run-to-run.

Unpredictable programming times lower the desirability uses-case of FGs. However, characterizing and extracting parameters for FG-dense platforms present a large initial overhead for designers. While accurate, there is a motivation for a solution that is predictable and does not require *apriori* knowledge. This technique is referred to as linearized programming [51]. Furthermore, this technique became the basis for continuous-time programming which is presented in the following section. In Fig. 6.2(b) an op-amp is applied to create a negative feedback loop from V_s to V_{cg}. Modulating V_{cg} will ensure a constant V_{sd} is held constant as well as a constant V_{fg} to V_s. Therefore, linearization creates the conditions for a constant (i.e. predictable) rate of injection. This is the exact technique used in [51] except it followed a pulsed programming methodology to find the programming stopping point. Finding a continuous-time programming stopping solution will be largely what distinguishes the design from its discrete-time counterpart.

Knowing the application space that the FG-employed circuit is going to operate in is another aspect a designer wants should consider before choosing a pulsed programmer. For example, will a circuit need the ability to program in-the-field once it is deployed to adapt operational parameters as its environment changes? Alternatively, will the FG-employed circuit largely inhabit a setting with benchtop equipment accessible? While not having been explicitly stated, pulsed based programming requires peripheral circuitry and equipment. The read phase will need some means of measuring the injection of the preceding injection pulse. Past arts have done this off-chip with a picoammeter, and a combination of on-chip transimpedance amplifiers with an off-chip 12-bit analog-to-digital converter [51, 53]. The programmers in [52, 53] will need storage for a look-up table (LUT) which may be in the form of an FPGA or microprocessor. Furthermore, there needs to be a means for controlling the highly precise pulse timing widths which can be in the tens of microseconds [52].

However, pulsed programming does have its benefits. If one has the proper equipment at their disposal, it's conceptually straight forward to implement injection partially or fully off-chip. Additionally, the read state utilizes the nominal supply voltage like the run state (i.e. when the FG is applied to a circuit), so the designer can be assured of very similar operation in the run state. Furthermore, the biggest benefit of pulsed programming is that it is supremely accurate given there is no limit on programming time because one could pulse with thin programming pulses for the finest control. [53] reported 9.5 bits of accuracy and [51] reported > 13 bits of accuracy.

6.3 Continuous-Time Programming

Continuous time programming is accomplished in a single operation as opposed to N precisely timed discrete programming pulses. This is shown qualitatively in Fig. 6.1(b). The means for achieving programming in a single operation is through negative feedback. Negative feedback can be used to (1) linearize the injection process (i.e. hold V_{fg} constant during

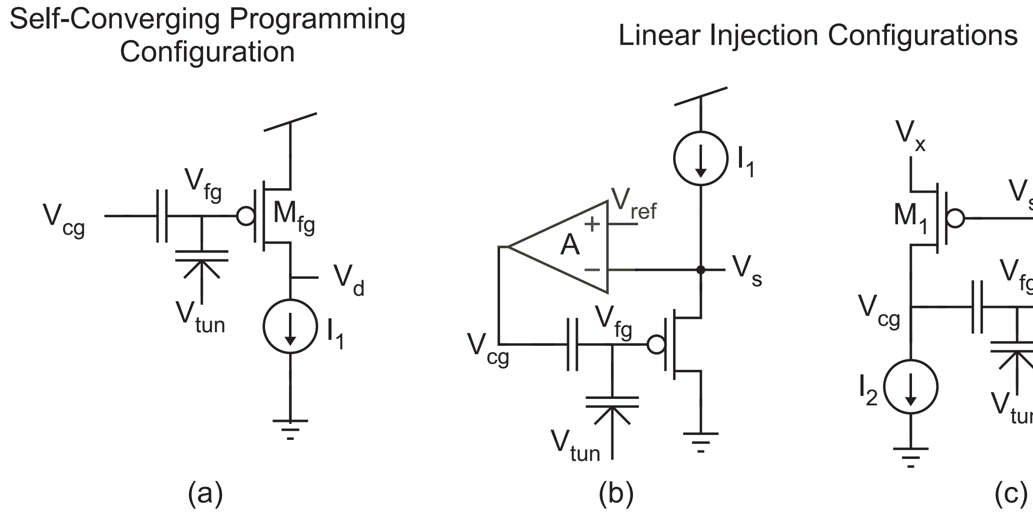


Figure 6.2: Continuous programmers. (a) Self-converging injection configuration. Lacks feedback for linearization, however, inherent feedback of the configuration stops injection once programming target is reached. (b) Operational amplifier holds V_s (and consequently V_{sd}) at a known value via V_{cg} which allows for a constant rate of injection. (c) Current-controlled current conveyor structure that allows for independent control of V_s and consequently V_{sd} via negative feedback through V_x and I_2.

injection) and (2) stop the injection process when the programming target is reached. These two operations are not always accomplished with a single feedback loop, and usually require at least two loops. Continuous-time programming will be type of injection implemented in this work. This work will serve as a comparative analysis of two different continuous typologies.

One of the most basic continuous-time programmers is simply a single FG configuration pictured in Fig. 6.2(a) [49]. With a favorable injection V_{sd} applied to an FG, electrons inject onto the floating-gate node, effectively lowering its potential. Referring back to (6.1), the net effect of reducing V_{fg} will lower I_s, and consequently lower I_{inj}. This configuration displays an inherent feedback to stop injection as the FG is programmed to its target. However, this process does not exhibit linearized injection (because V_{fg} changes) which would need an additional of a negative feedback loop.

The operation of 6.2(a) exhibits nonlinear changes in injection, which can lead to long convergence times since I_{inj} changes during programming. This lowers the desirability for using FGs since there can be wide ranging programming times. Recall from the discrete-time linearization technique using a closed-loop op-amp configuration [51] became the basis for continuous-time programming. This configuration is shown in 6.2(b), but lacks the ability to stop programming once a target has been reached. In this state, it's not much different from [51]. This circuit will be explored and compared against Fig. 6.2(c).

Lastly, Fig. 6.2(c) is another linear injection programming cell that creates similar conditions like that of Fig. 6.2(b) in a more compact size to be used in FG-dense structures like an FPAA. This particular configuration is referred to as a "current-controlled current conveyor" structure whose origins were originally described in [54], but first applied in programmer applications in [27]. V_{cg} is set by I_1 via common-source amplifier M_1, which is where its gets the "current-controlled" namesake. The current I_2 is "conveyed" to V_x. As a programmer circuit, this gives the designer the ability to set I_1 to an appropriate current that is favorable for injection to occur while appropriately adjusting V_{cg} to keep V_{fg} constant. At the same time, V_{fg} to V_s forms a source follower during programming. With V_{fg} constant, the source follower configuration allows for a constant V_{sd} during injection. All transistors operate in the subthreshold regime during injection, and therefore are subject to their subthreshold single transistor amplifier expressions.

Fig. 6.2(b) and Fig. 6.2(c) will be the two typologies explored in this work. The benefits of the topology in Fig. 6.2(b) promise to be more accurate and consistent in programming because 1) the op-amp can be designed with a much larger open-loop gain of than that of a common source amplifier and 2) V_s can be chosen via V_{ref} in a closed-loop configuration. In 6.2(c), the common source amplifier does not give the designer the ability to set a specific voltage at V_s; its capabilities are limited to setting the appropriate programming conditions allowing for linear injection.

The current-controlled current conveyor structure offers a more compact design and has been the traditional method of programming for past FPAA designs [4, 8]. Both FPAA designs contain hundreds of FG cells, and contained within each cell is the current-controlled conveyor. This is possible because it's a single transistor. However, this configuration would not be tenable for a full op-amp because of the amount of silicon space it would consume. The overhead of the the past designs of [4, 8] have not allowed for a single, all-encompassing linear injection programmer with a configuration of 6.2(b). Future programmer designs will be focused towards the integration of the op-amp for the benefit of removing the FG cell-to-cell mismatch stemming from the common-source amplifier.

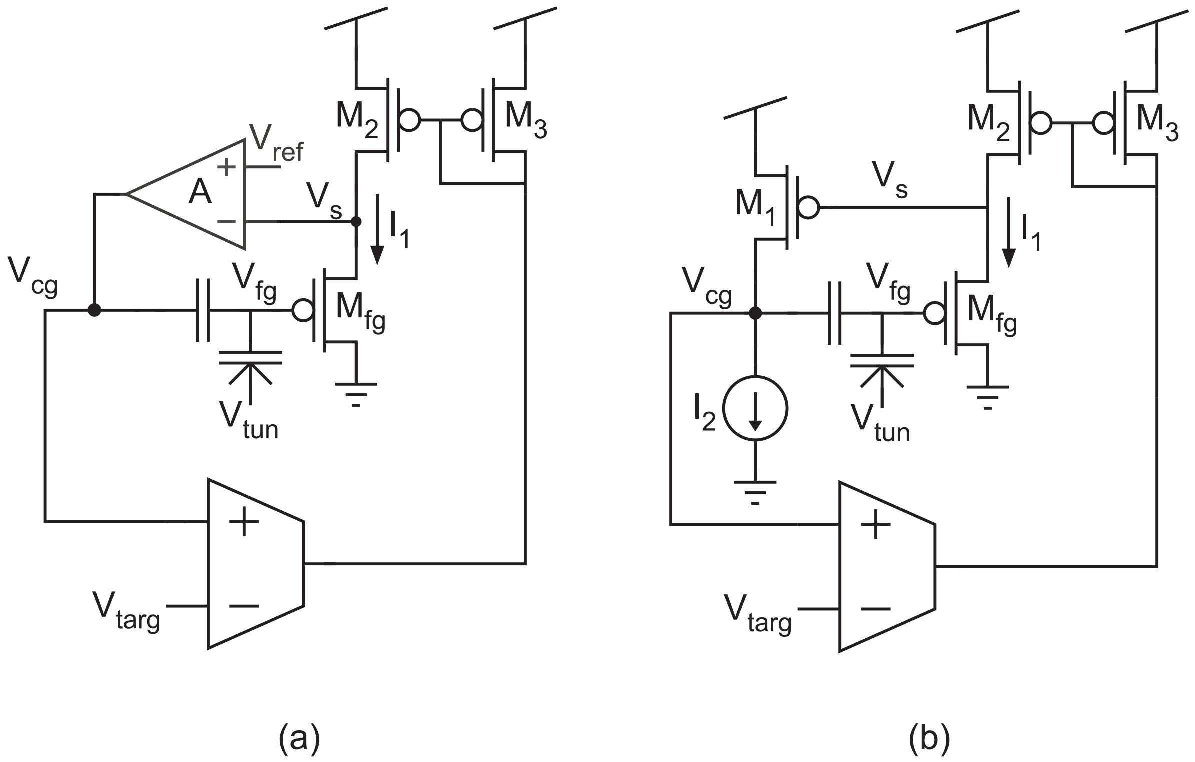


Figure 6.3: Continuous-time programmers with target converging circuitry that consists of an OTA and current mirror M_2/M_3. Target converging circuitry operates the same for both topologies of (a) and (b). During injection, V_{cg} increases linearly from V_{ss}. OTA compares V_{cg} to user-provided V_{targ}. When V_{cg} reaches V_{targ} trip-point, the OTA cutoffs I_1 drain current to effectively end the injection of electrons onto the FG.

6.3.1 Programming Target Convergence Structure

Additional support circuitry is needed for Fig. 6.2(b) and Fig. 6.2(c) to converge upon a desired programming target. This target will be in the form of a voltage and will be accessible to the designer. To realize this functionality, the circuit would need the ability to compare the programming target voltage to an additional node to monitor the injection progress of the FG. Once the target has been reached, the circuit would then need to end the injection process. These needs are addressed with the circuitry of Fig. 6.3 which includes the original continuous-time linear injection programmer of 6.2(b) and (c), and an operation transconductor feedback configuration.

The convergent circuitry of Fig. 6.3 is a feedback loop consisting of M_2, M_3, and the operational transconductance amplifier (OTA) for both linear injection programmers configurations. The OTA functions as the voltage comparator of programming target V_{targ} and V_{cg}. V_{cg} will increase linearly during injection and was discussed in the previous section. As

V_{cg} increases, the OTA will compare it against V_{targ} which will serve as the trip-point for when programming should end. As long as $V_{targ} > V_{cg}$, the OTA will set I_1, via the M_2-M_3 current mirror, to the OTA's tail current. When V_{targ} reaches the programming target V_{cg}, the OTA will set the M_2-M_3 current mirror in cutoff and I_1 will reduce to zero. This ends the injection programming of the FG since there is not any current that could impact ionize at the drain of the FG.

The described injection sequence is visually demonstrated in Fig. 6.4. The top plot displays V_{targ} and V_{cg} that are compared at the OTA. The bottom plot displays the FG drain current I_1. The initial conditions for this simulation assume that the the FG does not have any charge on it (i.e. all charge has been tunneled off). At the beginning of this sequence, the FG has not been enabled for injection, and consequently, the signals have the following conditions: V_{cg} is at ground (i.e logic LOW), V_{targ} is set by the designer at 1.5V, and I_1 is zero. The programming is commenced at 0.035 seconds, at which point we see the I_1 increase towards the tail current bias of the OTA at $900nA$. It takes another millisecond to discern any changes in V_{cg} as the addition of electrons on the FG have not made a significant effect.

For the time directly after 0.036 seconds, V_{cg} rises linearly and I_1 is completely saturated to the OTA tail current. At about 0.039 seconds, V_{cg} becomes approximately V_{targ}. At that point, V_{cg} is immediately pulled high to V_{dd} (i.e. logic HIGH) and arrests the injection programming. At the same time of V_{cg} being pulled HIGH, there is a precipitous drop in I_1's current to zero. This response is due to the OTA feedback loop which operates as follows. When $V_{targ} < V_{cg}$, the OTA output is pulled HIGH and places M_2, M_3 in cutoff. Common source amplifier M_1 no longer has a proper voltage bias for operating in the saturation region, and consequently, cannot operate as an amplifier to offset the recently injected negative charge on the FG node (i.e. by holding V_{fg} constant). This forces V_s to be pulled LOW via the FG. As a result, M_1 is set in the triode region and pulls V_{cg} HIGH. V_{cg} is held HIGH for the remainder of the time that the FG is enabled for programming.

A short time after 0.04 seconds, there is a cessation of the injection programming process as V_{cg} is pulled LOW; the FG has been de-selected and there is no longer a V_{sd} favorable for injection applied to the FG. At this point and for the rest of the graph, the signals are restored to the original conditions before injection programming commenced. This concludes the injection programming process for a single FG. In a FG-dense design like FPAAs, this process is typically serialized as only one FG can be programmed at a time. The designer can

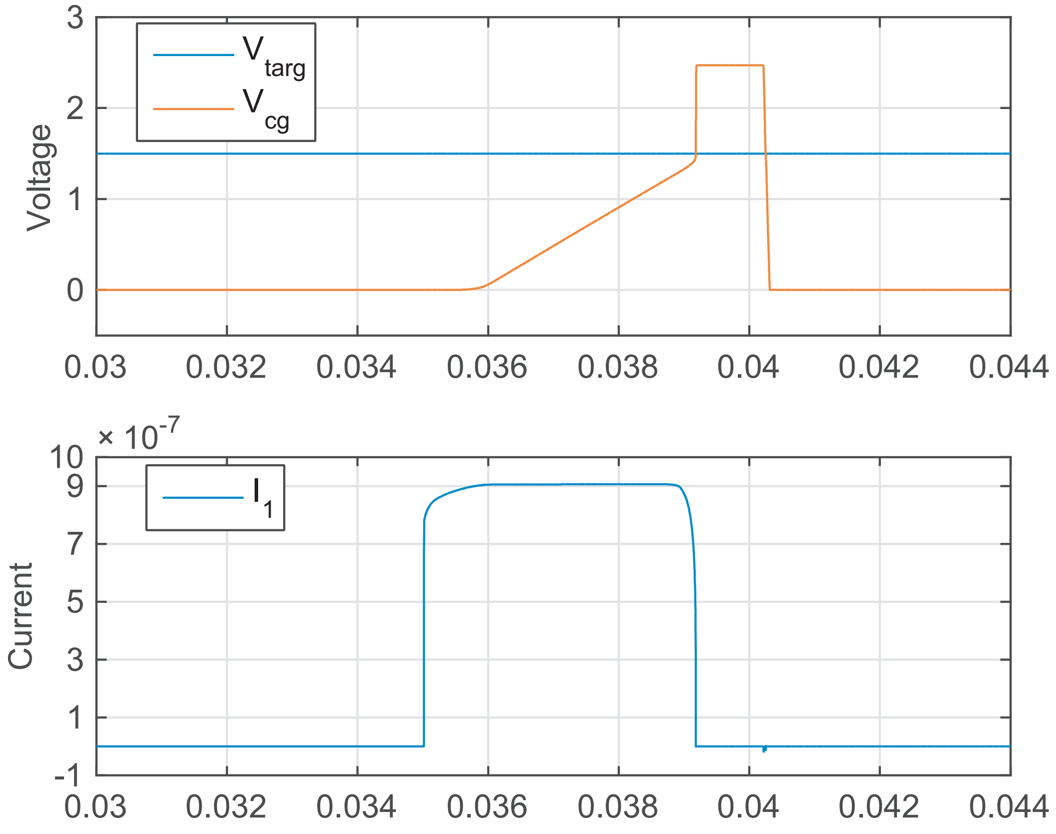


Figure 6.4: Transient simulation of the injection process utilizing the continuous-time linear injection programmer with target converging circuitry of Fig. 6.3(b). The plots demonstrate that during injection, FG drain current I_1 is saturated to an appropriate current conducive to injecting electrons. In the above case $I_1 = 900$nA. Once V_{cg} reaches V_{targ}, FG drain current I_1 drops precipitously to end the injection programming process.

then subsequently choose to inject a different FG which may have a different programming target voltage from the preceding FG.

The continuous-time programmer with target converging circuitry of Fig. 6.3(a) operates the same as Fig. 6.3(b) since the underlying circuitry is fundamentally the same structure. The only difference is that the op-amp has differential input while the common-source amplifier has a single input. In fact, the graph of Fig. 6.4 could have been generated from either circuit. Just like Fig. 6.3(b), the OTA of Fig. 6.3(a) places M_2, M_3 in cutoff when V_{cg} reaches V_{targ}. V_s is then pulled LOW by the FG, and consequently, the op-amp then operates as a comparator and pulls V_{cg} HIGH. This shows that the programming target convergent circuitry operates independently of amplifier choice and allows for a comparative study on the two programmer topologies. The outcome is to determine which programmer programs FGs more consistently and accurately. Before the two topologies can be directly compared, an additional aspect needs to be explored: above-ground and below-ground programming.

6.4 V_{sd} Voltage for Injection

Up to this point, V_{sd} favorable for injection has been characterized in Section 5.4.2 as a voltage applied across the FG source-to-drain that is typically larger than the nominal voltage supply of the process, but less than the junction breakdown voltage. At that point in the discussion, it was unnecessary to detail the specifications of a favorable V_{sd} for injection. However, it is germane to this section regarding FG injection circuitry and it will be elaborated on here.

As described in section 5.4.2, for injection programming occur, one of the conditions is to have a large V_{sd} (i.e. $V_{sd} > V_{dd}$) potential; this potential may be positive or negative voltage. A positive V_{sd} pulls the source above V_{dd} and has been the primary choice in many past works focused on FG-dense structures [2, 4, 27, 55, 56, 57, 58].

A negative V_{sd} is created by pulling V_s of the FG below-ground. This is less intuitive since all other on-chip circuits utilize a single, positive supply voltage. There are a number of benefits that could be gained if the injection voltage was implemented below-ground compared to above-ground. Some of the major benefits include less overhead infrastructure, and the promise of better injection accuracy. The main issue for employing below-ground injection programming has been implementing it in a standard, single-well CMOS process because of the potential of forward biasing diffusion areas. Above-ground and below-ground injection programming are compared in this section.

6.4.1 Above-Ground Injection Voltage and Associated Programming Circuitry

Applying a positive V_{sd} voltage is the most conventional way of injecting an FG. These duties are typically performed by a charge-pump and the signal is applied in the form of a pulse whose duration is long enough to complete the injection programming. The signal is not sustained indefinitely because of the long-term stress of a voltage drop exceeding voltage supply rail of the process. Furthermore, for above-ground programming, FGV_{dd} will refer to a separate supply voltage used by all programming-associated circuitry, including the FGs. In run-mode/circuit-mode, $FGV_{dd} = V_{dd}$. During programming, FGV_{dd} will be equal to the charge-pump output voltage. It will become clear why this distinction is needed.

In previous contexts, the discussion regarding injection was based solely from the perspective of the quantum mechanical concept. In this context, the discussion is concerned with how injection is carried out with circuit infrastructure for the application of FG-dense structures like FPAAs. Furthermore, it should be determined what circuitry is involved beyond the programmer and what circuitry should be isolated from this process. The amount of circuitry that experiences the large V_{sd} pulse should be minimized.

Any circuitry that interfaces with the programmer or FG needs its voltage supply to be level shifted to FGV_{dd}. In fact, their voltage supply should be connected to a separate supply from V_{dd} referred to as FGV_{dd}. This includes all digital control signalling and bias circuitry needs of both the programmer and the FG. But, before examining the interconnection of these circuits, the FGV_{dd} signal should be characterized. In Fig. 6.5(a), the charge pump signal used to generate FGV_{dd} is shown. This signal poses a number of challenges for circuits that will rely on it as a voltage supply.

One of the first issues that the charge pump is not in operation when there is no programming needed (i.e. run-mode or circuit-mode). The output of the charge pump when not enabled will be 0V. Therefore, the circuits would need another source to supply $FGV_{dd} = 2.5$V for run-mode operation. For this problem, there is a need for a facilitator circuit to choose its supply and be capable to switch between a programming voltage supply (charge-pump) and a typical supply voltage ($V_{dd} = 2.5$V). This circuit must be able to handle seamlessly switching between the two sources and not be affected by the charge pump's long time-constant when decaying from 6.5V. This long time-constant is showing in Fig. 6.5(a).

The proposed circuit would have functionality like that shown in Fig. 6.5(b), where FGV_{dd} would be the supply voltage seen by the level-shifted circuits. Beginning at 0 seconds, when the charge-pump is not enabled and outputting 0V, $FGV_{dd} = 2.5$V. This would be typical of run-mode operation. When programming is commenced at 1.25 seconds, control signal V_{select} changes state to set $FGV_{dd} = V_{charge-pump} = 6.5$V. Once programming has finished, control signal V_{select} changes back to its original state to select $FGV_{dd} = 2.5$V at 2.75 seconds. $FGV_{dd} = 2.5$V regardless of $V_{charge-pump}$'s state. These measurements were completed using the on-chip high-side switch of Chapter 3 and on-chip charge-pump. The time span shown in Fig. 6.5 is not typical of the programming sequence for a FG as it's

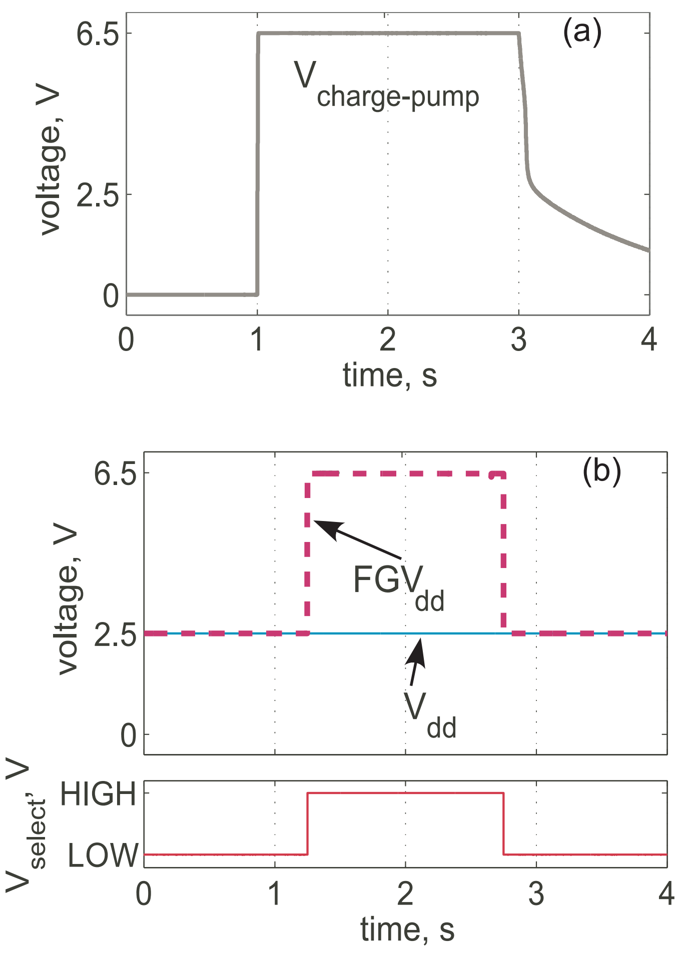


Figure 6.5: Functionality of a supply-voltage selecting circuit. (a) Typical charge-pump signal used for programming FGs. The ramp-up time is on the order of microseconds and the discharge time is on the order of milliseconds to seconds. (b) Shows the functionality of a voltage supply selecting circuit. The dashed maroon signal FGV_{dd} selects and outputs higher voltages when programming (i.e. charge-pump is enabled). While run-mode, it is outputting the nominal supply voltage V_{dd} regardless of $V_{charge-pump}$'s state.

usually occurs in tens to low one-hundreds of milliseconds. The time span of the figures are for conceptual demonstrative purposes.

A circuit that could implement the functionality Fig. 6.5 is a high-side load switch whose design is fully elaborated in Chapter 3. A block diagram that demonstrates the voltage source switching functionality is shown in Fig. 6.6. Its design incorporates pFET switches that allow it to operate without a secondary charge-pump, which is typical of nFET-based high side switches. Additionally, it meets a criterion of FG programming in its ability to switch to supply voltages larger than the nominal supply voltage. The output of the high-side load switch would be connected to the supply of the aforementioned circuits.

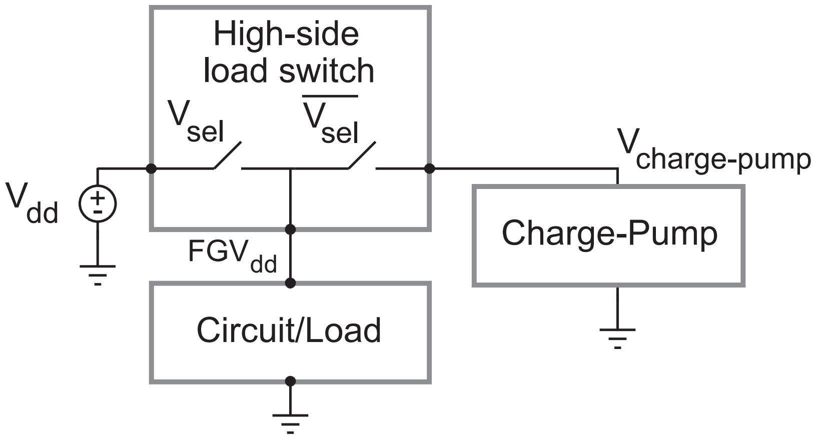


Figure 6.6: High-side switch block diagram. This circuit is capable of choosing one of two supplies by a signal control signal. It's design is unique in that it can select a supply voltage larger than its nominal supply voltage without a secondary charge-pump.

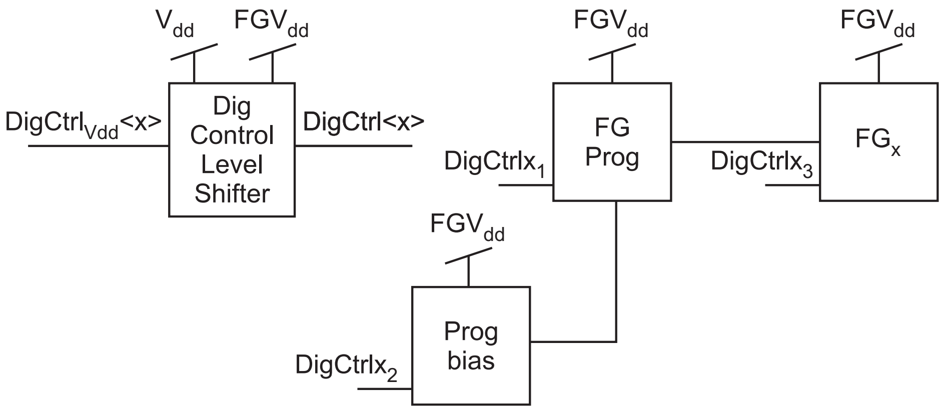


Figure 6.7: Block diagram of all the circuits in our FPAA design which rely on the high-side load switch and their relation among each other.

The circuits that rely upon the high-side switch's FGV_{dd} are shown in Fig. 6.7. The analog circuits include FG programmer (FG Prog), the selected FG (FG_x), and FG programmer bias (Prog bias). The digital control signals need to be translated from 2.5V to FGV_{dd}. There is a level-shifter for each digital control signal. When taken together, implementing an above-ground programming scheme has some overhead associated with it in terms of circuitry and control (e.g V_{select} of the high-side switch).

In summary, above-ground injection programming is the most conventional methodology of programming FGs. Recently developed circuit high-side switch circuit has reduced the overhead associated with above-ground injection by integrating it on-chip. There are additional overheads of above-ground programming to account for including voltage level shifters and pad frame repercussions.

Above-Ground FG Cell Operational Modes

Figure 6.8(a) shows the schematic of an individual FG cell of an FG array like the one of [4]. Fig. 6.8(b) is a table of the various operational modes and their respective connections for a FG cell. The first two rows refer to injection programming configurations. The first row is the selected FG when connected in the programmer (in gray) feedback loop, while all other FGs of the array are disconnected. The unselected FG configuration shown in the second row does not experience the large V_{sd} voltage as M_1 is OFF such that $V_{sd} = 0$V. Therefore, this is the selection process in which only the selected FG is injected. “RUN-mode” and “FG Measure” rows refer to configurations in which FGs are connected internally or externally, respectively. These three configurations feature external control of the V_{cg} node and are user-accessible. The RUN-mode configuration allows all FGs to be connected internally on-chip to circuits, while the FG measure configurations refer to a single FG being connected off-chip. The FG measured selected configuration is utilized for test purposes and debugging. Both V_{cg} and S_{w4} position 3 are global connections to their own respective pins.

6.4.2 Below-Ground Injection Voltage

As its name implies, below-ground programming achieves a large V_{sd} conducive for injection by pulsing V_d below 0V. This is not a purely academic proof-of-concept. Implementing below-ground programming on-chip for FG-dense structures has a number of impediments, however, there are many benefits to be gained by choosing this course. The most obvious benefit of below-ground programming is obviating the infrastructure associated with handling FGV_{dd} circuits like level-shifters and a high-side load switch. For this case, every circuit shares one and the same V_{dd}. Less infrastructural overhead also results in lower power consumption.

There are larger and more consequential benefits that impact the programming process when compared to above-ground programming. Recall that in above-ground injection, the source voltage of the FG is ramped up above V_{dd}. Raising V_s obviously affects the V_{sg} voltage which is one of the mechanisms that controls the channel current. Not only is it an inconvenience that V_g must be modified and ramped up proportionally with V_s, but more importantly, V_s must be a stable voltage for accurate programming. V_s is generated by a charge-pump and any ripple on V_s modulates the V_{sg} voltage which harms the accuracy of the programming. Choosing below-ground pulsed injection is also beneficial to the common-source amplifier Fig. 6.3 (b) since it has a relatively low open-loop gain and will theoretically have less ripple on the source to compensate for during programming.

Since above-ground injection programming pulses the source, this means below-ground pulses the drain. A V_d transient does not affect the V_{sg} voltage and its affect on the channel current is much less pronounced in sub-threshold saturation than a V_s transient. Overall, choosing a below-ground programming regime can further reduce performance requirements of the charge-pump for injection. Any voltage ripple from the charge-pump's output now occurs at a node with a much larger impedance and is therefore less coupled to the drain current. A direct result of this is the charge-pump voltage-ripple tolerance can be lowered. Another benefit is that the below-ground charge-pump does not need to generate as large of a voltage magnitude to achieve injection. For example if a charge-pump pulses a FG V_d to -4V, the net V_{sd} from a $V_{dd} = 2.5$V is 6.5V. For an above-ground charge-pump to achieve the same V_{sg}, it does not get the benefit of $V_{dd} = 2.5$V contributing to the overall V_{sd}, and must generate a full 6.5V.

FG Selection and Isolation for Below-Ground Injection Programming

The benefits of below-ground programming are very apparent, but there have been impediments for implementing the circuit infrastructure to accomplish it in a standard single-well CMOS process. Specifically, the largest issue has been carrying out below-ground injection programming on a single, selected FG cell among a full array of FGs. This section will examine the why traditional selection methods that worked for above-ground, do not work for below-ground programming. Finally, a solution will be presented to solve this problem.

On an FG array, there is a single charge-pump shared among all FGs via a global source (drain) connection for above-ground (below-ground) programming. In order to isolate the selected FG cell for programming from the others, the traditional methodology (i.e. used in above-ground) is to employ transmission gates or simply a FET switch for this purpose. This is shown in Fig. 6.8(a) where S_{w3} chooses the source voltage. However, employing a complementary technique at the global drain for below-ground injection becomes an issue. The n-type diffusion regions of the transmission gate forms a parasitic pn-junction to the p-type substrate. And when the charge-pump output goes negative, a forward bias occurs.

Furthermore, utilizing just pFETs as transistor switches could not accomplish passing a negative voltage without additional circuitry to operate the transistor in the triode region. Therefore a new selection/isolation process for below-ground programming is in required.

The solution to achieve this FG selection functionality is to utilize a matched, secondary pFET that interfaces with and shares the FG node. This is pictured in Fig. 6.9(a) and was originally demonstrated in [59]. In effect, this configuration obviates transmission gates, and instead, each pFET of the FG is delegated to one of the two separate tasks: programming and circuit-mode operation. When in injection programming mode, the 'Circuit pFET' is put in a dormant state by holding the source and drain at the same potential – in this case ground. This does not permit the flow of current through $M_{circuit}$ and does not affect injection operations.

When configured in an injection programming mode, the selected (unselected) FG in the array has a potential from V_{dd} (V_{ss}) to the below-ground charge-pump output voltage. This configuration is accomplished by the 'Injection pFET' multiplexer which selects position 2 (position 3) via S_{w4}. For the selected FG in the array, the charge-pump is still able to create a favorable source-to-drain voltage for injection to occur. All the other deselected FGs of the array do not experience injection since the potential from ground to the negative charge-pump output voltage is not large enough for this phenomenon to occur. Furthermore, the unselected FG cells are taken out of the programmer's feedback loop via switch S_{w3}. These two injection configurations are shown in the first two rows of Fig. 6.9(b).

Below-ground FG cells operate similarly to above-ground FG cells during circuit modes (i.e. RUN-mode and FG measure). The external, user-controlled voltage is passed into V_{cg}, and the drain of $M_{circuit}$ is permitted to be connected either internally or externally – depending upon the code-word. The internal connection is used to connect all the FG cells to their respective circuits which are used to synthesize sensing circuits on the FPAA. The external configuration is used primarily to testing and debugging. For all circuit-mode configurations, M_{inj} has to be set in such a way that it does not affect the circuit-mode operation of the FG cell. Since the charge-pump is not in use during this operation and it's output is pulled HIGH to 2.5V, the source is pulled to HIGH. This does not permit the flow of channel current through M_{inj} and does not circuit-mode operations. All the various programming and circuit-mode configurations are shown in Fig. 6.9.

6.5 Programming Accuracy

A comparison of the various programming configurations needs to be made in order to determine the most accurate configurations. However, before this may occur, a methodology is needed to compare the results in a fair, one-to-one way. The most ideal situation would be to measure the amount of charge placed on a floating-gate after injection. Unfortunately, there is not a way to probe the floating-gate, and it's too sensitive of a node to probe. An alternative method of accomplishing the same outcome would be attaching a buffer to the floating-gate, much like the same way a circuit transistor was simply added to the below-ground programming cell. However, this would take up a lot of silicon space, and furthermore, from FG-to-FG and chip-to-chip there is associated mismatch with transistor that could lead to inaccurate comparisons.

Another problem in carrying out a comparison is that each programming configuration will achieve a different outcome. Even between the below-ground cases of Fig. 6.3, having the same bias current for the OTA and selecting the same V_{targ} will not achieve the same outcomes because of the fundamental differences within the topology. While the gate-to-source amplifier with the smaller gain could lead to less accurate results but not fundamentally different IV curves, the different V_s's will lead to different IV curves. In Fig. 6.3(b), there is no way of setting V_s, and no way of knowing V_s unless simulated or buffered out to pin. Fig. 6.3(a), however, keeps V_s constant at a user-provided value. Both of these V_{sd} voltages are set in different ways and will lead to differences in programming FGs. The issue is further compounded when comparing a below-ground result with an above-ground result.

The solution to comparing the different programming topologies cases is actually quite simple. Instead of comparing in terms of charge change on a floating-gate buffered out as a voltage, the comparison should take place in the current-domain. This especially makes sense considering our target application of FGs will be in the form of programmable current sources. More specifically, this methodology should show the variation in current among n experiments and should show the equivalent variation in terms of V_{cg}. In other words, to achieve our desired current at a likely percent-error, how much variance in V_{cg} can be expected? Knowing the variance in V_{cg} is useful because all FGs on an array will operate with a single V_{cg}, therefore it's advantageous to know how precise a V_{cg} should be in order to achieve an equivalent output currents – especially for currents biasing sensitive circuits.

The manner in which this solution is carried out can be done by mapping all results back to a “reference curve”. The reference curve is an arbitrary current-versus-V_{cg} plot that was measured from a programmed FG. The output current from the FG needs to span multiple magnitudes of ten in order to capture the whole range of currents that could possibly be mapped. The only constraint to this is that the range of currents can only be achieved within the supply rails $0V < V_{cg} < V_{dd}$. Because an arbitrary reference curve can be created by any FG injection topology, and in fact, one cannot discern from which topology the FG was programmed, we can map all data points back onto the reference curve to have a one-to-one comparison.

Fig. 6.10(b) gives a visual example on how this methodology is carried out. The full, blue IV-curve is the reference curve and the left three dots are the measurements from three repeated programming experiments under the same conditions. In this context, a programming experiment is defined as a completely tunneled FG, injected to a specific V_{targ} under the conditions of a known V_{sd}, and programmer OTA bias current. Each programming trial is repeated exactly the same. After injection, a specific V_{cg} is applied and the injected FG’s output current is measured. This V_{cg} voltage has been characterized and determined before the experiment. As shown in the Fig. 6.10(a), the three trials are measured at the characterized V_{cg} voltage, which under ideal conditions should give a known (i.e. anticipated) output current.

In this example of Fig. 6.10(a), the current should be about $110nA$, but the three different measurements shows that there is some variance in the programming. At the known V_{cg} voltage, all three corresponding current measurements are mapped back to the equivalent current value on the reference curve. Once the values are mapped to the reference curve, the points are projected on both the y-axis and x-axis. This is shown in Fig. 6.10(b). From the y-axis, the standard deviation and mean of the currents can be calculated. From the projected data on the x-axis, the standard deviation of V_{cg} can be calculated. The overall percent error of current can be calculated from the y-axis measurements as I_σ/I_μ. With these three statistical metrics, a user can know the overall percent error of the expected target current. This can then correspond to an equivalent millivolt deviation from the ideal V_{cg} in order to obtain the expected current.

Not only is this methodology useful in comparing inter-programmer results (i.e. below-ground programmer with op-amp versus above-ground), but also intra-programmer results. For example, the below-ground programmer with a common-source feedback amplifier might operate most accurately with an OTA bias current of $800nA$ compared to a bias at $2\mu A$. An intra-programmer comparison can determine such cases as well.

A table of measurements utilizing this methodology are shown in Table 6.1. The table displays data from the common-source amplifier configuration and operational amplifier configurations, as well as above-ground and below-ground topologies. Each row represents the experimental conditions and each statistical category was based on 100 experiments. An explanation of the column headers are outlined in the following.

OTA bias (μA)	The bias current provided to to the programmer OTA. Refer to Fig. 6.3 OTA programmer;
V_{inj} (V)	Above-ground or below-ground voltage used for injection;
V_{targ} (V)	The user-supplied target voltage for injection. This is an input to the programmer OTA and can be referenced at Fig; 6.3
Feedback Amplifier Type	Common-source amplifier (CS-amp.) or operational amplifier (op-amp) along with a specified V_s voltage. Refer to Fig. 6.3 for both configurations;
$I_{measurment}$ % Error	The measured current percent error from 100 mapped experiment iterations for the specified target current at the preceding conditions;
V_{cg} standard deviation (mV)	The calculated V_{cg} voltage standard deviation calculated from 100 mapped experiment iterations for the specified target current at the preceding conditions;
10μA - 1nA	These represent the five target currents that span multiple magnitudes of ten. After the FG has been programmed, five known V_{cg}s are applied that correspond to these specific currents. The corresponding currents measured out from the FG are the values that are mapped to the reference curve which are then used for the statistical metrics

One of the more general patterns gleaned from the measurements is the error increases as the target current decreases. This trend makes sense given that the change in target current is an exponential change and the operation at the sub-1μA are in the subthreshold region. This is not the result of less preciseness in the programming because the V_{cg} standard deviation stays relatively constant throughout all the experiments under all target current

Table 6.1: Re-programmability measurements from a below-ground and above-ground topologies. Both utilize either common-source amplifier (CS-amp) configuration and operational amplifier (op-amp) for closed-loop linear output. Each row of data represents an experimental case that was repeated 100 times.

Case Conditions				**$I_{measurement}$ % Error, V_{cg} standard deviation (mV)**				
OTA bias (μA)	**V_{inj} (V)**	**V_{targ} (V)**	**Feedback Amplifier Type**	**$10\mu A$**	**$1\mu A$**	**$100nA$**	**$10nA$**	**$1nA$**
0.8	-3.5	0.60	CS-amp.	0.193%, 0.839	0.574%, 0.869	1.16%, 0.838	1.59%, 0.816	1.72%, 0.826
0.8	-3.5	0.85	CS-amp.	0.193%, .840	0.578%, 0.873	1.16%, 0.838	1.61%, 0.825	1.72%, 0.830
0.8	-4.0	0.60	CS-amp.	0.190%, 0.818	0.548%, 0.831	1.10%, 0.803	1.53%, 0.791	1.65%, 0.807
0.8	-4.0	0.85	CS-amp.	0.158%, 0.684	0.477%, 0.715	0.964%, 0.700	1.35%, 0.699	1.44%, 0.697
0.25	-3.4	0.80	op-amp, V_s=2.0V	0.217%, 0.939	0.651%, 0.942	1.30%, 0.949	1.79%, 0.916	1.95%, 0.901
0.25	-3.4	1.4	op-amp, V_s=2.0V	0.245%, 1.03	0.764%, 1.11	1.51%, 1.10	2.10%, 1.07	2.34%, 1.08
0.25	-4.0	0.80	op-amp, V_s=2.0V	0.227%, 0.957	0.696%, 1.01	1.40%, 1.01	1.97%, 1.00	2.15%, 0.996
0.25	-4.0	1.4	op-amp, V_s=2.0V	0.253%, 1.06	0.786%, 1.13	1.56%, 1.12	2.16%, 1.10	2.37%, 1.10
2.0	+5.5	3.75	CS-amp	0.087%, 0.385	0.244%, 0.342	0.485%, 0.351	0.632%, 0.330	0.813%, 0.362
2.0	+5.5	4.75	CS-amp	0.100%, 0.431	0.279%, 0.397	0.548%, 0.385	0.708%, 0.355	0.926%, 0.374
2.0	+6.5	3.75	CS-amp	0.134%, 0.586	0.353%, 0.498	0.375%, 0.484	0.845%, 0.436	0.971%, 0.472
2.0	+6.5	4.75	CS-amp	0.087%, 0.376	0.255%, 0.364	0.513%, 0.358	0.666%, 0.333	0.884%, 0.349

cases. $1nA$ is simply 10,000 times smaller current than $10\mu A$ and is further compounded from this current being in the subthreshold region. That is, for a small linear change (i.e. δ <10mV) in the applied V_{cg}, there is an exponential change in the output current in this region. Stated a different way, for a smaller percent error deviation from the programming target with exactly the same applied V_{cg}, there is an exponential change in the output current. On the other hand, $10\mu A$ is an above-threshold current which is in the above-threshold saturation region. Therefore, for a small linear change in V_{cg} results in relatively unchanged output current.

A discernible difference between the data sets is the op-amp style programmer effectively has double the $I_{measurement}$ percent error and V_{cg} standard deviation; both sets are <4% and <2mV, respectively. I posit that the increased error/standard deviation is due to the relatively low open-loop gain in the source-to-gate op-amp. The structure of the op-amp is a two-stage type with about the same gain ($\approx$40) as the common-source amplifier. It is further constrained in that it must maintain V_s=2.0V for all test conditions during operation. The op-amp programmer does not have the benefit of utilizing the full V_{sd} range from V_{dd} down to the V_{BlwGnd} voltage like the common-source amplifier. Through testing and schematic simulation, it was found that for V_s<2.0V, one or more of the transistors would fall out of saturation region. V_s=2.0V represented a good balance between a high V_s so that a charge-pump was not required to below -4.0V. For a future design, I postulate that increasing the gain of the op-amp significantly higher and sizing the transistors internally that would allow for V_s>2.0V would lead to better or the same results as the common-source amplifier. Overall this configuration is preferred because the V_{sd} can be known to the designer which will lead to better FG programming characterization.

Other factors to consider that are not apparent in the table is injection programming duration, which can contribute to an unwanted second-order effect called reverse-tunneling. The OTA bias current is the current that gets mirrored into the FG during injection. From eq. (6.1), it is clear that the channel current affects the rate of injection. Therefore, the smaller channel currents, the longer it will take to meet the programming target with all other factors being the same. For smaller currents like 250nA and under, this is on the order of 120-300ms. For larger currents like 800nA and above, this on the order of 30-85ms. Its possible to see reverse-tunneling effects after long periods (on the order of seconds) of injection programming, which could be possible if one is programming multiple FGs on a FPAA-like structure. The pronounced effect is seen as unwanted injection of non-selected

FGs; when a large V_{sd} is present for long periods of time, this can tunnel electrons onto all of the unselected FGs. Hence, this is an aspect to consider when choosing the best programming conditions.

An alternative and more graphical method of analyzing the re-programmability data is to create a linearization plot like the one shown in Fig. 6.11. For this particular plot, an above-ground programmer programs a FG 100 times at each V_{targ} voltage within a range; the range shown here is the complete operational V_{targ} range for this above-ground programmer. Once programmed, a V_{cg} voltage is modulated on the control gate until a desired output current is measured — in this case, $10nA$ is the desired current. For the top plot of Fig. 6.11, each circle represents the mean applied V_{cg} from 100 experiments that achieved an output current of $10nA$.

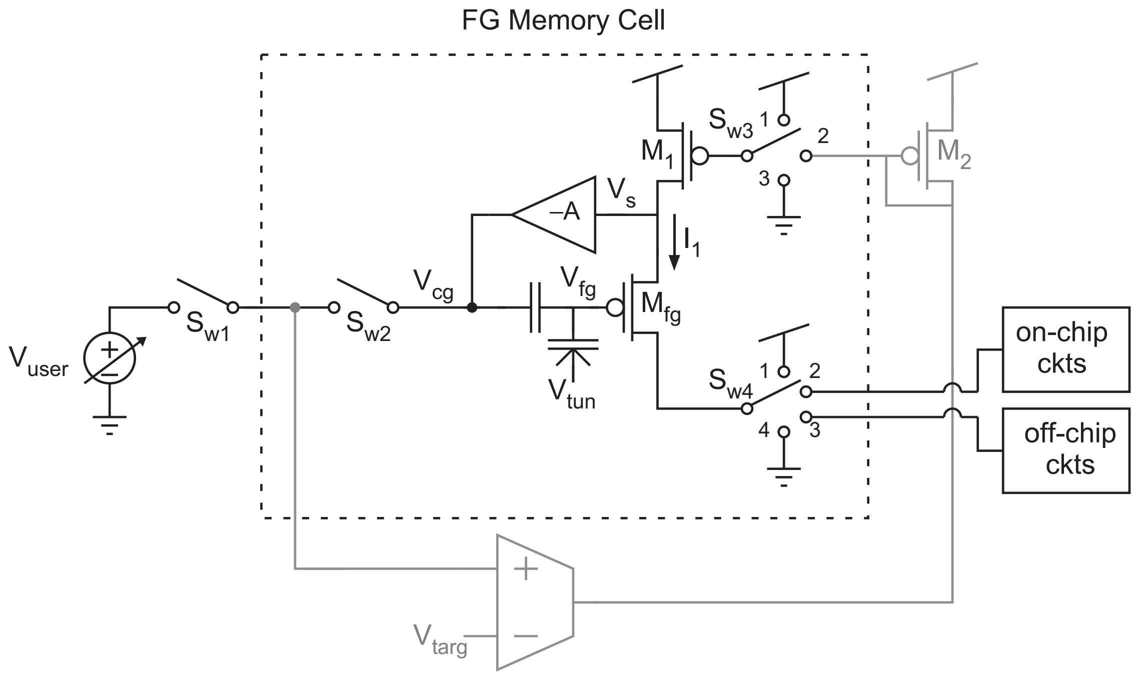
The bottom plot of Fig. 6.11 shows the deviation of each the mean V_{cg} (i.e. blue circles) from the linear regression of the top plot's line. The linear regression has a slope of 0.995. The line being linear with nearly a gain of 1 confirms our earlier discussion regarding the closed-loop operation of the continuous time programmer. The candlesticks shows standard deviation for each 100 programming experiments.

Finally, the motivation for characterizing an FG programmer is not to analyze metrics, but is to consistently and accurately program FGs for their end-use applications. And the best way to demonstrate this is utilizing them in a typical application. Pictured in Fig. 6.12, are the frequency responses of eight bandpass filters at octave, half-octave, and third-octave spacings, respectively. Each filter has its -3dB corner frequencies set by FG biases. In total, there are 16 FGs biases built within an array, and because they are re-programmable, the end-user has the option to choose the octave spacing among each filter. The filters shown in Fig. 6.12 were created via Capacitance Coupled Current Conveyor whose corner frequencies are a function of their internal transconductances, which are set by the programmed FGs.

6.6 Conclusion

Hot-electron injection programming is the preferred method of adding charge to FGs over Fowler-Nordheim tunneling for FG-dense structures like FPAAs. This is due to the ability to selectively program single FGs without disturbing the charge states of other FGs on the same chip. There are two conventional methods of carrying out injection programming: pulsed programming and continuous programming. For this work, continuous programmers were explored because these designs are conducive to a completely on-chip solution that would benefit IoT applications and in-the-field changes.

Below-ground and above-ground continuous-time programmers were described for comparative study on accurately repeatable programming. The comparative study methodology was created such that different programmers could be analyzed on the same basis. Two below-ground continuous-time programmers were characterized and measured. The common-source amplifier type had slightly better results compared to the op-amp type below-ground programmer, but this can be attributed to a sub-optimal op-amp design. Their utility was demonstrated in programming various octave spacing bandpass filters with an array of 16 floating gates. The above-ground continuous-time programmer has been characterized and measured for this work as well.

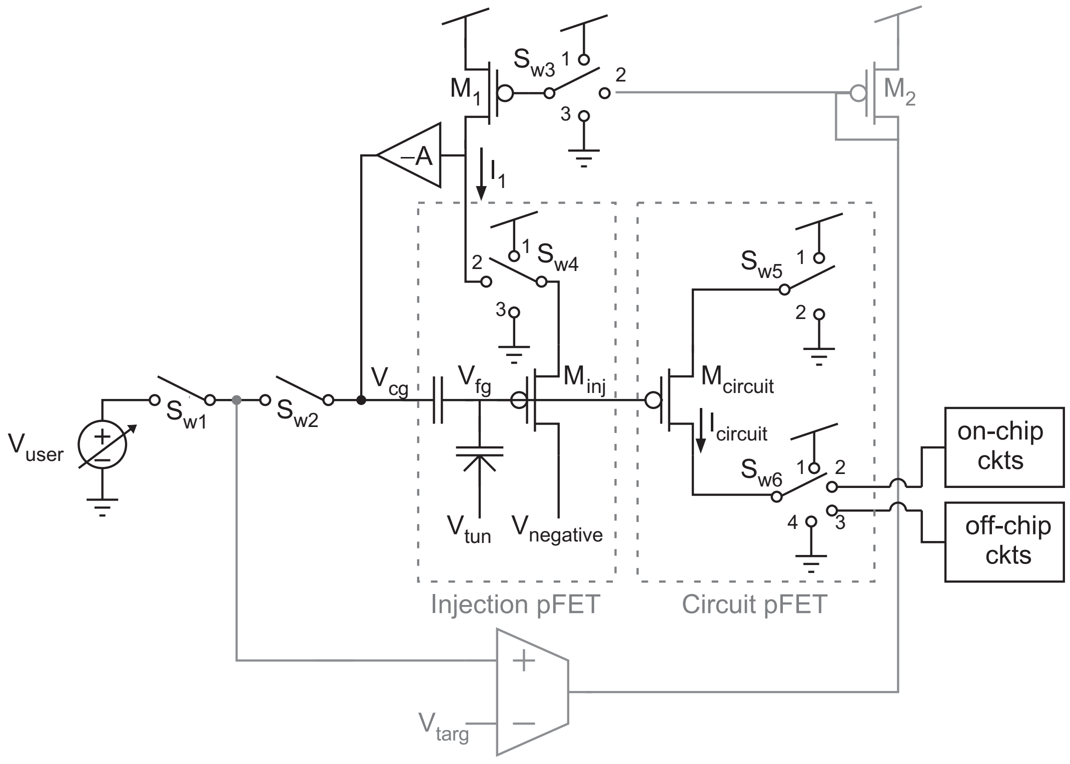


(a)

	S_{w1}	S_{w2}	S_{w3}	S_{w4}
Injection selected FG	open	closed	pos. 2	pos. 4
Injection unselected FG	open	open	pos. 1	pos. 4
RUN-mode	closed	closed	pos. 3	pos. 2
FG measure selected	closed	closed	pos. 3	pos. 3
FG measure unselected	closed	closed	pos. 3	pos. 1

(b)

Figure 6.8: Above-ground FG memory cell various operational modes and their respective connections. In (b) the first two rows refer to programming modes in which the selected FG is connected to the programmer, while all other FGs of the array are disconnected. RUN-mode refers to a configuration in which programmed FGs are connected to internal circuits. FG measure selected/unselected refers to a configuration in which a single FG from the array is connected off-chip.

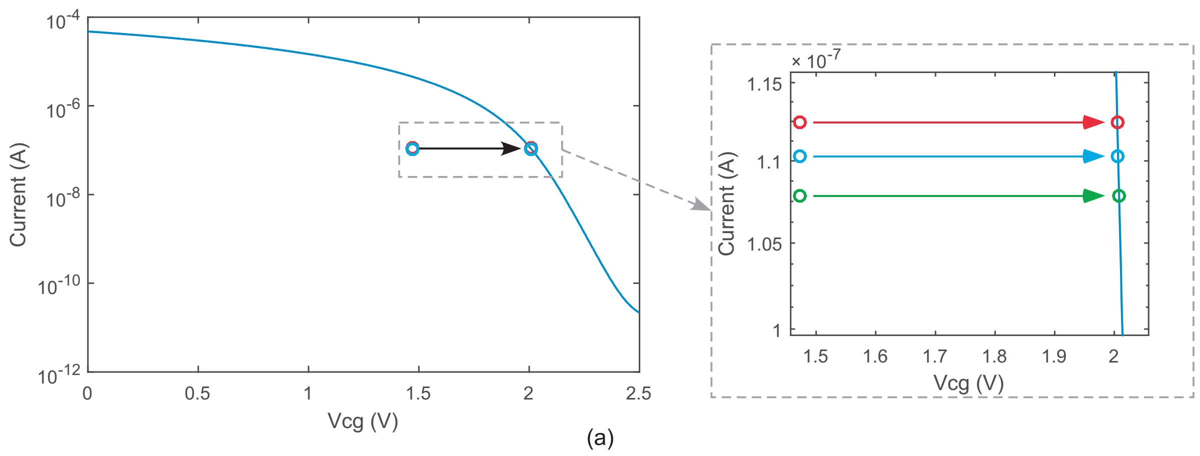


(a)

	S_{w1}	S_{w2}	S_{w3}	S_{w4}	S_{w5}	S_{w6}
Injection selected FG	open	closed	pos. 2	pos. 2	pos. 2	pos. 4
Injection unselected FG	open	open	pos. 1	pos. 3	pos. 2	pos. 4
RUN-mode	closed	closed	pos. 1	pos. 1	pos. 1	pos. 2
FG measure selected	closed	closed	pos. 3	pos. 1	pos. 1	pos. 3
FG measure unselected	closed	closed	pos. 3	pos. 1	pos. 1	pos. 1

(b)

Figure 6.9: Below-ground programming FG circuit configurations for injection and circuit modes (i.e. RUN-mode and FG measure). Both memory cells allow for FG selection/deselection in an arrayed FG configuration such that only the desired FG is programmed. Both configurations allow for circuit-mode operations.



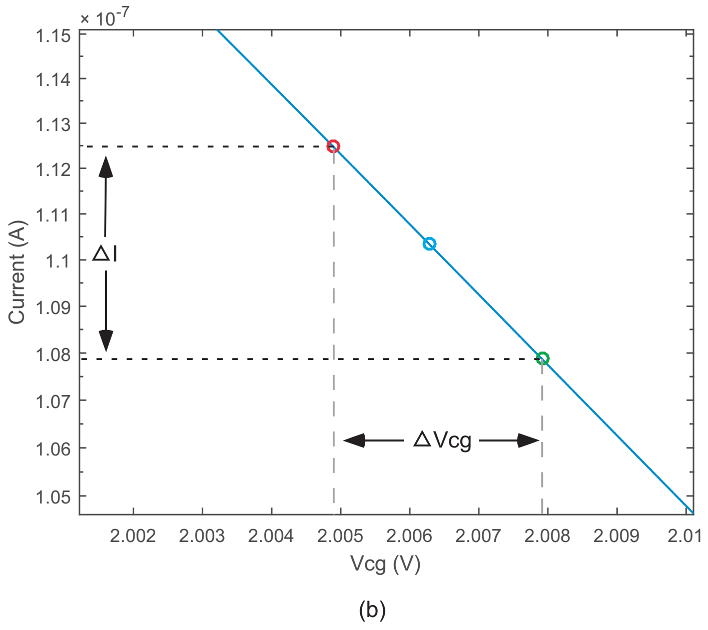


Figure 6.10: (a) Programming experiments are mapped to a reference IV-curve to allow for one-to-one comparisons among different programming experiments. (b) Once the n number of trails have been mapped – in the above case 3 –, the data can be projected onto the x-axis and y-axis statistically calculate the accuracy of the programming experiments.

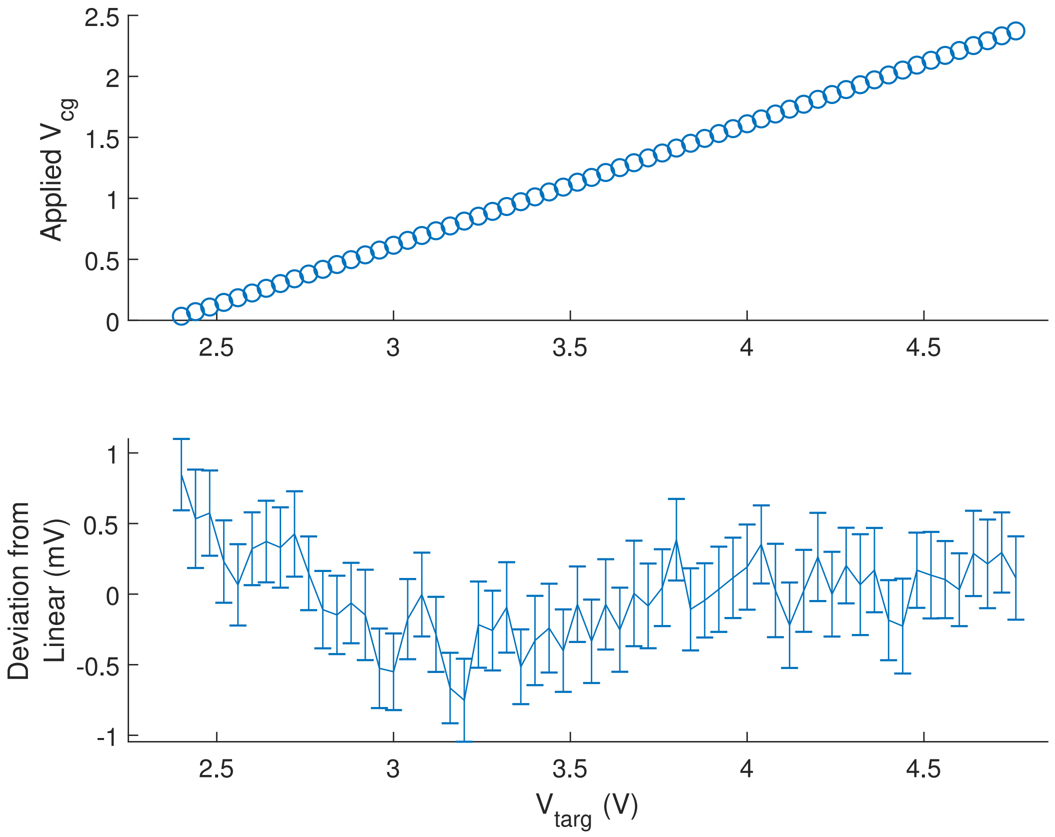


Figure 6.11: Linearization plot for an above-ground programmer. An FG is programmed to V_{targ} voltages in linear steps. After programming to each V_{targ}, the control voltage is modulated until the desired current is measured — in this case $10n$A. The continuous-time programmer operates in a closed-loop configuration and forces the linearization of programming. The bottom plot shows the deviation of V_{targ} from the linear regression line and the candlesticks shows the standard deviation for the 100 programming experiments.

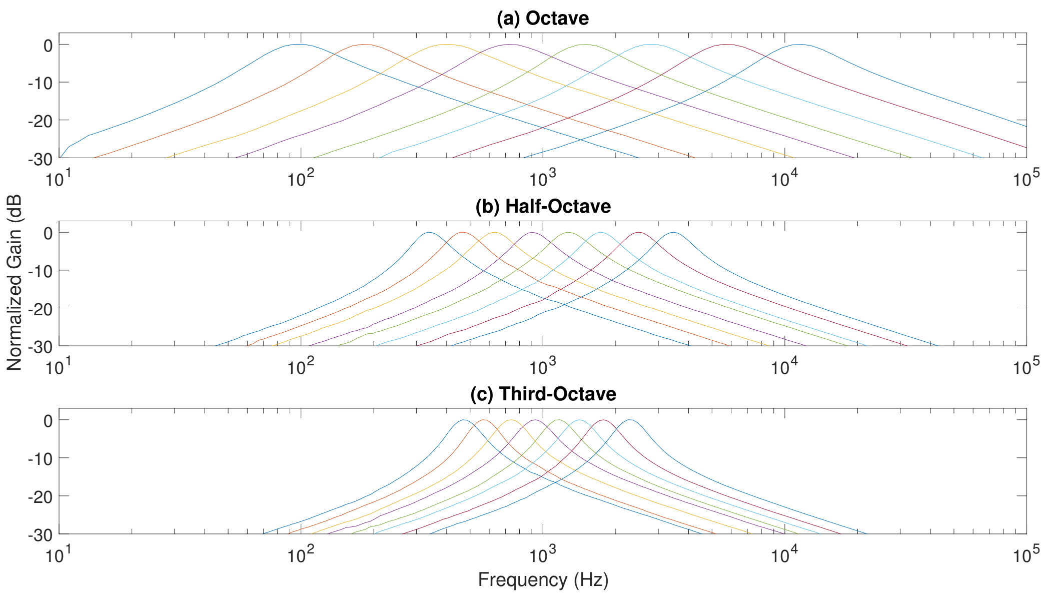


Figure 6.12: Programmed C4 bandpass filters at full, half, and quarter octaves.

Chapter 7

Reconfigurable Analog Preprocessing for Efficient Asynchronous Analog-to-Digital Conversion

Much of the previous chapters that have dealt with infrastructure-specific FG designs and not necessarily application development. On the other hand, the non-reconfigurable designs such as the work done in the DOE project were more system-level. This chapter bridges these fractured elements to present a unique end-user application for FPAA designs interfaced with an ADC. The reconfigurable analog-front-end design is employed to asynchronously sample information-specific waveforms.

7.1 Introduction

Whether the application is bio-signal monitoring with wearable devices, voice-activity detection by an Internet-of-Things device, or data logging with wireless-sensor-network nodes, modern electronics require application-specific data capture with as low a power budget as possible. The means that are used to sample and preprocess (usually compress) a signal comprise a major part of this power budget. Traditional analog-to-digital converters have made great strides in power efficiency, but their sampling rate is constrained by the Nyquist–Shannon sampling theorem. This rigid sampling-rate criteria can create a large amount of sample overhead in signals with varying frequency content, which manifests as extraneous power expenditure in the quantization, processing, and transmission stages.

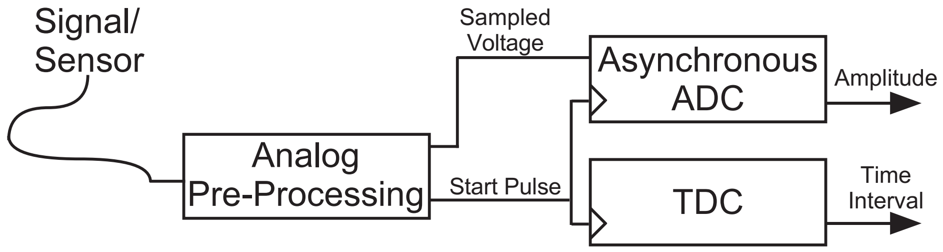


Figure 7.1: Asynchronous analog-to-digital conversion system. A reconfigurable analog front-end reduces information to only the relevant data points and also triggers the subsequent blocks, which produce digital words for the corresponding voltages and time intervals.

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, the demonstrated system asynchronously produces two digital signals—one proportional to the sample amplitude and the other proportional to the intersample time period, as shown in Figure 7.1. 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 [4]. The use of a reconfigurable front-end allows us to create a sample triggering mechanism tuned to the minimal amount of required data for a given application. The well-tuned analog front-end, combined with asynchronous data conversion, allows for highly efficient data extraction. This work is an extension of our earlier work presented in [60], which demonstrated the promise of an asynchronous ADC coupled with analog preprocessing. Here, we provide further circuit details, a discussion on a sampling method that works in conjunction with asynchronous conversion, and additional example applications.

This work demonstrates the use of a fabricated RAMP and an asynchronous ADC to convert the minimal number of samples required to either recreate a signal or optimally extract specific data from a signal. An overview of asynchronous conversion is presented in Section 7.2. Section 7.3 establishes the overall system architecture that we use to implement our adaptive sampling technique, including a discussion of our reconfigurable analog front-end and the data conversion circuits. Section 7.4 describes the asynchronous triggering process in this conversion system that allows for only the local maxima/minima to be sampled. Section 7.5 demonstrates example uses of the system in voice, electromyography (EMG), and electrocardiogram (ECG) monitoring environments, before concluding in Section 7.6.

7.2 Asynchronous Quantization

The most traditional path to achieving a reduction in energy expenditure of electronic devices is through the scaling of the minimum feature size of the transistors, which primarily benefits digital systems. Unfortunately, the conversion of an analog signal into a digital one is an inherently mixed-signal process. The analog portions of this process tend to suffer as transistor dimensions scale, due to the increased device mismatch and the lower voltage overhead (from reduced supply voltages). While there are techniques meant to mitigate these issues [61], process-scaling is usually not the simplest or most cost-effective means of reducing energy expenditure in the quantization stage.

The next direct path toward reducing the power expended on sampling a signal is to reduce the number of samples taken. The Walden figure of merit for analog-to-digital converters (ADCs) suggests that, for ADCs of comparable resolution, a reduction in power can be achieved through an equal reduction in sampling rate [62]. Unfortunately, traditional Nyquist–Shannon sampling sets the sampling rate at a hard minimum of twice the highest frequency content of a signal. While fixed-rate sampling has the advantage of being very-well understood and characterized, it does not provide an efficient means of sampling sparse or 'bursty' signals—signals characterized by relatively short periods of high activity and potentially long periods of inactivity (e.g., voice-control systems).

The extraneous quantization of signals creates increased energy overhead outside of the quantization stage as well. For these quantized signals to be useful, they must normally be processed by a microcontroller and/or transmitted to some other device by a transceiver. Logically, the more samples that are taken, the more samples that must be processed and transmitted by the microcontroller/transceiver. This has a compounding effect on the power consumption, as these activities reduce the amount of time it may stay in a low-power 'sleep' state, which operates at orders-of-magnitude lower power expenditure than the active state. For context, the active states of two low-power industry-standard components, the MSP430 microcontroller and the CC110L transceiver, consume over 200 times and 80,000 times more current than their respective sleep states [11]. It is, therefore, very desirable to leave these devices in a sleep-state as much as possible by reducing the sample overhead.

The desire to vary a data converter's sampling rate so that it may adapt to the changing frequency characteristics of a signal has given rise to data converters that rely on asynchronous sampling methods. Instead of wasting a significant amount of energy converting parts of the signal that contain relatively little data, asynchronous sampling methods base their rate on the signal itself. The trigger for these methods can be a number of different signal-characterization or classification circuits, but many asynchronous sampling solutions struggle with a robustness/resolution trade-off.

Level-crossing ADCs are possibly the most popular type of asynchronous ADCs [63, 64, 65, 66]. The basic concept 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 [67]. While this method avoids extraneous conversions during periods of low-activity, level-crossing ADCs have a severe resolution–bandwidth trade-off—the higher the resolution of the ADC, the longer it takes for large-amplitude high-frequency content to traverse through the many quantization levels.

Alternative triggering schemes can be made from analog front-ends designed for different signal-conditioning and classification schemes. For example, one could imagine an ADC triggered upon the detection of a certain frequency within a signal [56] or upon the detection of vocal characteristics of a signal [58]. Indeed, many energy-efficient analog front-ends [68] could provide a trigger for an asynchronous data converter. Unfortunately, these analog circuits are very application-specific, requiring new circuits to be designed on a per-application basis.

The desire to enable reconfigurability in analog electronics has led to the development of a relatively new class of devices called field-programmable analog arrays (FPAAs) [4, 69, 70, 71]. 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. Allowing circuit designers to reconfigure and retune analog elements and biases allow these FPAAs to function across a myriad of applications. In this work, we leverage our reconfigurable analog front-end—the RAMP—to provide user-defined triggers to enable conversion based upon the characteristics of the signal, thereby minimizing the number of conversions to significantly save power.

7.3 System Overview

When sampling signals with sparse information, constant-rate sampling unavoidably creates extraneous samples during periods of low activity. These extraneous samples create waste in the quantization stage as well as further down the signal-processing chain in the microcontroller and transceiver stages. The use of analog preprocessing has been shown to be computationally efficient to find the data points needing to be converted [68]. While a fully custom front-end would yield the highest energy savings, the use of a field-programmable analog array allows us to adapt the front-end for a variety of applications.

Once the FPAA-synthesized analog front-end detects some predefined characteristic of the signal, it calls for the immediate quantization of the signal. Due to the asynchronous nature of the quantization, not only must the amplitude be recorded, but the time interval since the last sampling event must also be recorded, as shown in Figure 7.1.

The amplitude conversion within the demonstrated system occurs with a simple successive-approximation analog-to-digital converter. From a system perspective, the unique aspect of this ADC is that it must be triggered to initiate conversion. This trigger signal also activates a time-to-digital converter (TDC). A TDC is essentially an oscillator with a digital counter attached to the output that counts the number of oscillations between extrema occurrences. By 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 intersample time measurement. The constant oscillation of the TDC may seem inefficient, but a unique feature of this system is that the ADC uses the TDC as its required clock signal. By replacing the system clock, the TDC offsets some of its own power budget within the system.

7.3.1 Reconfigurable Analog Mixed-Signal Platform

The reconfigurable analog/mixed-signal platform (RAMP) refers to our FPAA, which is capable of synthesizing large and complex circuits for event-driven and signal-processing designs. Coupled with a custom netlist-based language and place-and-route routines, the RAMP enables users without circuit-level expertise to quickly develop applications in silicon [4].

As shown in Figure 7.2, the RAMP contains 80 computational analog blocks (CABs) that are interconnected via ten stages with eight channels. The ten stages are organized by processing type, such as basic circuit elements (e.g., transistors and capacitors), transconductors, continuous-time filters, mixed-signal operations, etc. The eight identical channels

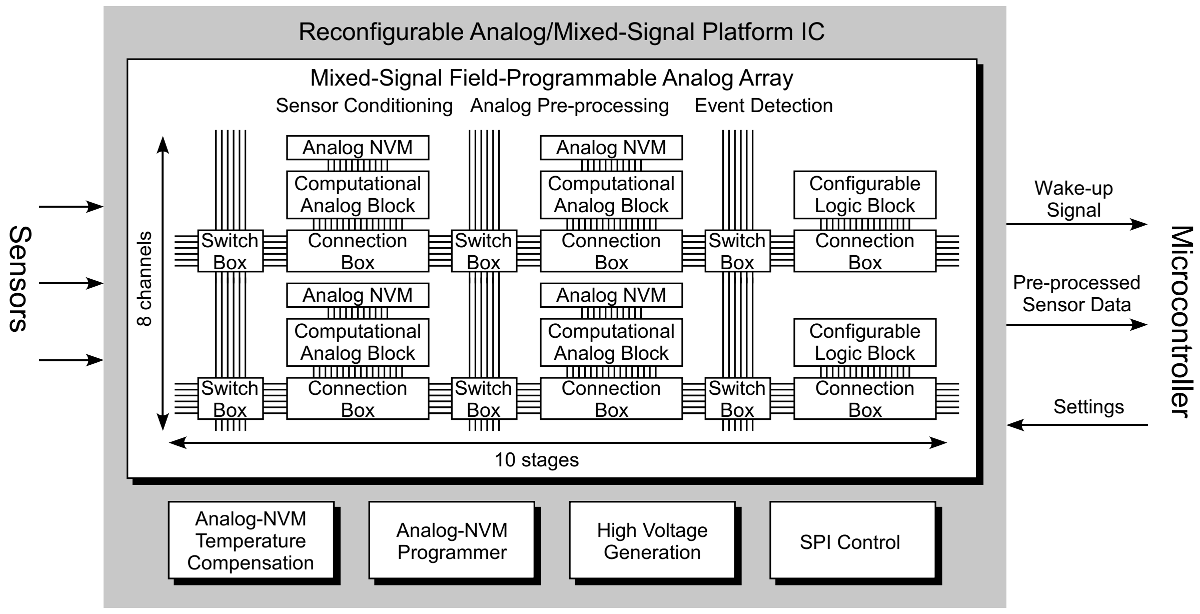


Figure 7.2: Architecture of the reconfigurable analog/mixed-signal platform (RAMP) integrated circuit.

provide the ability to create parallelized designs, as needed. Moreover, many of the synthesizable elements are tunable to set desired performance parameters (e.g., gain, filter corner frequency, etc.) and to tailor the respective circuit to the input signal characteristics. This tunable functionality is provided by floating-gate transistors, which are described in more detail in [27].

7.3.2 Asynchronous Data Converter Design

The voltage amplitude conversion of the asynchronously sampled values is performed by a successive-approximation ADC (SA-ADC). The successive-approximation architecture was chosen for its demonstrated power efficiency [72]. 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. The following discussion is devoted to the specific circuits shown in Figure 7.3 that give this ADC its functionality including the successive-approximation register, comparator, and time-to-digital converter.

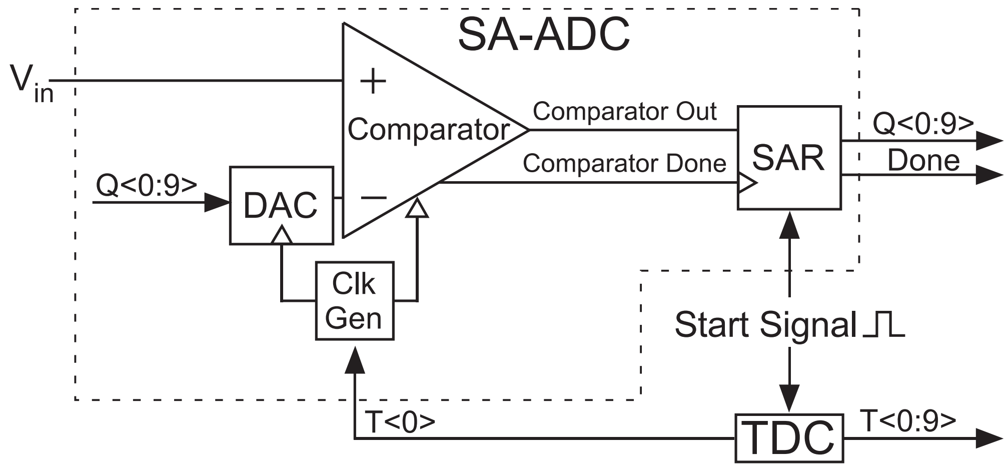


Figure 7.3: System diagram of the successive-approximation ADC (SA-ADC)/time-to-digital converter (TDC). A trigger from the RAMP analog preprocessing stage acts as the 'Start Signal' to initiate conversion and record the time.

Successive-Approximation Register

The successive-approximation register (SAR), which is shown in Figure 7.4, 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 Figure 7.4 is simply a shift register. This portion of the SAR passes a logic high down the chain of flip-flops upon each clock cycle, which is provided by the 'Comparator Done' signal, as shown in Figure 7.3. When a flip-flop in the shift register outputs a logic high, a corresponding flip-flop below it is also set high. This bottom row of flip-flops applies the digital word to the digital-to-analog converter (DAC) that provides the voltage to which the input is compared. 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—this event signifies that the SAR has finished all of its cycles, and thus the analog-to-digital conversion is complete. The SAR implemented in this work is 10 bits.

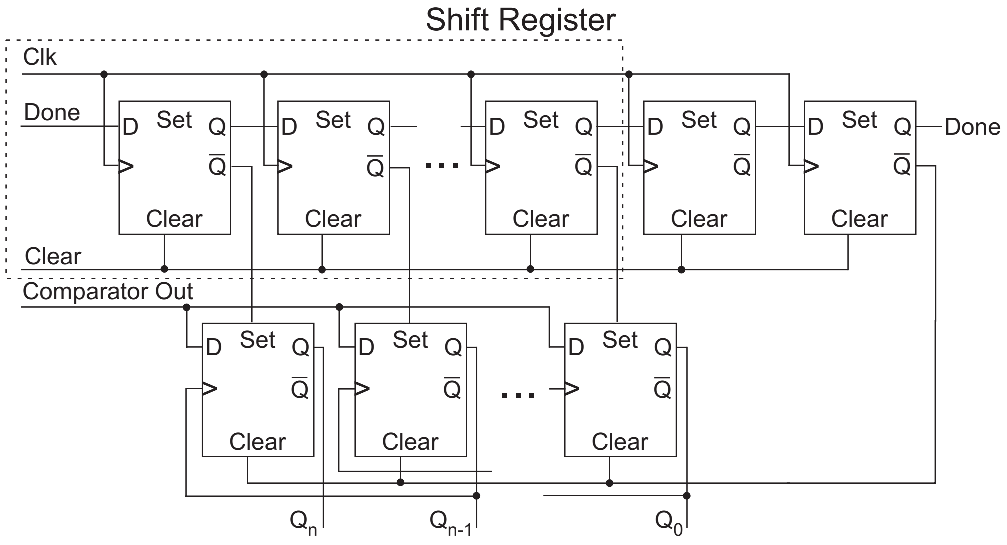


Figure 7.4: A schematic of the successive approximation register. The boxed area acts as a shift register. The lower set of D flip-flops is where the bits are actually applied to the rest of the system and stored appropriately.

Comparator

Maximizing resources within energy-constrained systems is a high priority, but it can be difficult, for example, to convert voltages along the entire full-scale range. The comparator shown in Figure 7.5 was based on the design in [73] 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 because the DAC output will be selected to be midrail by the SAR. 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, 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. The appropriate comparator is chosen by the MSB of the SAR (shown as En

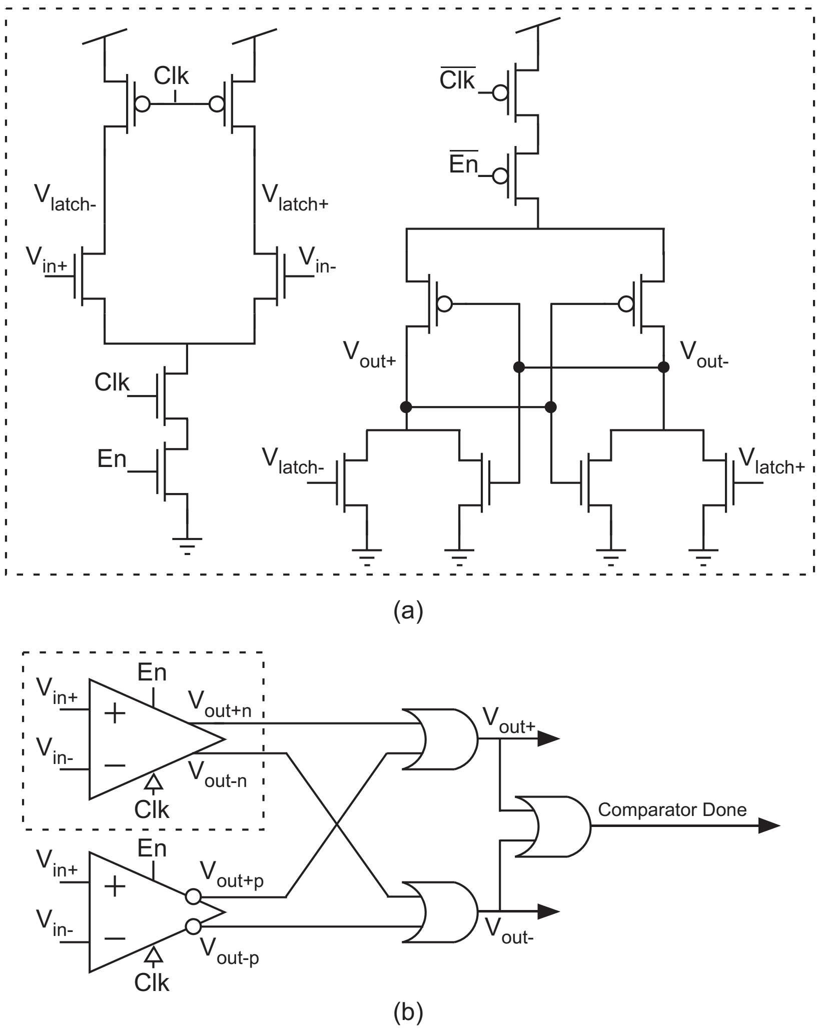


Figure 7.5: (**a**) Comparator schematic of the nFET-based comparator. A complementary pFET-based comparator is also used in parallel. (**b**) Complete comparator schematic showing both the nFET-based (dashed box) and pFET-based comparators operating together. The MSB selects which version to use to maximize the input range of the complete comparator circuit.

in Figure 7.5), thus ensuring that the comparator never operates in a region where it cannot make an accurate comparison. By intelligently selecting which comparator is used, we are able to convert values along the entire full-scale range.

Our clocked comparator is designed to use negligible static energy by breaking the connection between V_{DD} and ground when the clock is low, and also very little dynamic energy by utilizing minimally sized transistors. Prior to comparison, the clock signal is low, which precharges the V_{latch} nodes. When the clock signal goes high, the V_{latch} nodes are discharged at rates proportional to the voltages at 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. Figure 7.5b shows the interaction of the nFET- and pFET-based comparators. Their outputs are combined through logic OR gates to provide the final output as well as to create a comparator done signal that serves as the clock signal for the subsequent SAR block.

The drawback to using two symmetric comparators is that two different input-referred offsets must be accounted for in postprocessing. For this implementation, no layout matching techniques were used to address this unequal offset, but it would be a viable technique for helping to mitigate 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 are matched [74]. Alternately, by using floating-gate transistors as the input transistors, the comparator could be reduced to a single, non-symmetric version (i.e., as shown in Figure 7.5a) by programming the common-mode charge of the two floating-gate transistors to remove offset as well as to cover rail-to-rail input voltages.

Time-to-Digital Converter

The intersample times are recorded using a time-to-digital converter (TDC). A TDC is simply a device that takes some periodic signal and uses it to estimate the time interval between signal pulses [75]. Our TDC implementation, which is shown in Figure 7.6, was designed to take a small amount of physical area and digital support circuitry. The periodic signal is created by current-starved inverters arranged in a voltage-controlled oscillator topology. The current starving inverter, which is shown in Figure 7.7a, ensures that we can both mitigate the power wasted by short-circuit current and that we can control the

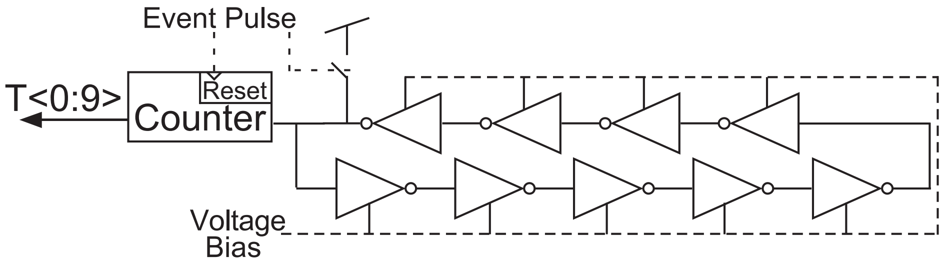


Figure 7.6: Time-to-digital converter consisting of a voltage-controlled oscillator and a counter. The counter keeps track of the number of oscillations. The event pulse clears the counter and also shorts part of the oscillator to logic high, which ensures that it begins in the same state after every reset.

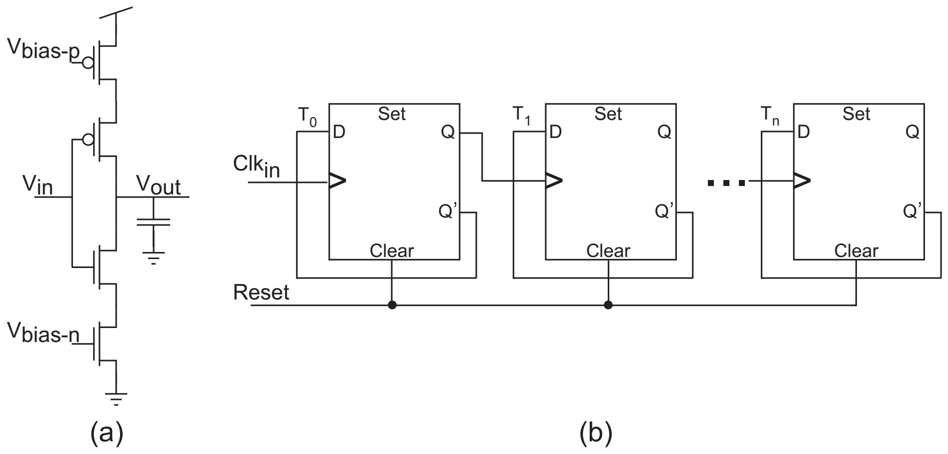


Figure 7.7: (**a**) A simple delay unit using a current-starved inverter. The delay is proportional to the size of the capacitor and inversely proportional to the current flowing through the inverter. (**b**) A simple binary counter. The D flip-flops are sequentially linked and output successive binary numbers.

frequency of oscillation of the overall device. The voltage bias is held at a DC value and the output oscillations are recorded using a digital counter. The output of this counter is then a binary word that is proportional to the length of time since the last restart pulse. The restart/event pulse is provided by the RAMP system and is used to reset the counter values to zero when a new sample is detected. The current version of this system does not account for overflow conditions of the counter, but future iterations would trigger a conversion by the ADC when the TDC reaches a state of all ones in order to maintain accurate representation of very-low-frequency signals.

7.4 Asynchronous Sampling Implementation

7.4.1 Extrema Sampling

The first task in implementing this system is to determine what signal quality will serve as the ‘trigger’ for conversion. In conventional data-conversion systems, a fixed sampling rate that is greater than the Nyquist sampling rate is used to acquire the samples—therefore, the system clock is the only external trigger needed for acquiring new sample values for conventional ADCs. On the other hand, asynchronous sampling methods adapt the sampling rate to the changing frequency content of the signal, so triggers within the signal itself must be used to determine when to take a new sample.

The RAMP device could be used to implement a variety circuits capable of finding triggers embedded within the signal that can be used to initiate conversion. We have chosen to focus on the detection of local extrema (i.e., local maxima and minima) to implement an ’extrema sampling’ system [66, 76]. In extrema sampling, the local minimum/maximum voltages are converted when the first derivative of the input is zero. Because the sampling is asynchronous, the time between each sample must also be recorded.

It is also worth reiterating that adaptive-sampling techniques, such as extrema sampling, are a good opportunity for power savings in converting ‘bursty’ signals—ones that have short segments of high-frequency events followed by large segments of relative inactivity. No energy is wasted on digitizing the input signal if it is not changing.

Figure 7.8 illustrates the reduction in the number of samples that can be obtained over traditional fixed-rate sampling techniques. The highest-frequency spectral component defines the Nyquist sampling rate for fixed-rate sampling systems—for this example waveform, 27 samples would be required over this time period when sampling at the Nyquist rate. In extrema sampling, only the local maxima and minima are sampled. In this way, extrema sampling adapts itself to the changing frequency content of a signal. When applied to this example waveform, only 9 sample values are required to accurately represent the same waveform. The reduction in the number of sampling points when using extrema sampling becomes more pronounced during periods of inactivity and/or only low-frequency content. As a result, significant energy can be saved through the conversion process, and superfluous data points are never converted.

The original signal can be reconstructed from the extrema samples by employing a simple method used for constructing complex lines in computer graphics called Bézier curves [77].

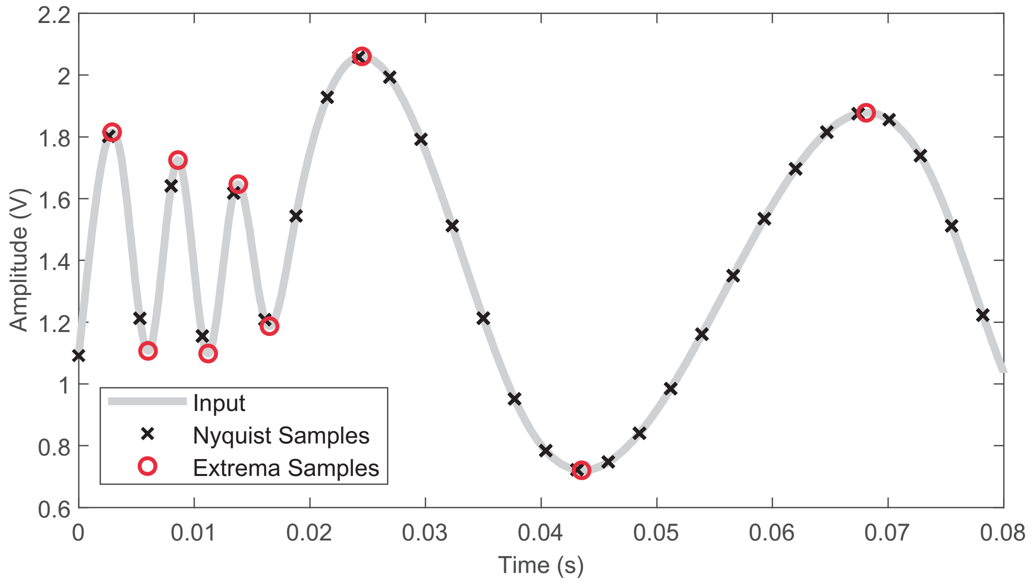


Figure 7.8: Comparison of constant-rate Nyquist sampling versus an adaptive-sampling method using extrema sampling.

The Bézier curve is a cubic polynomial capable of providing a smooth curve between two points when given well-defined inflection points. The cubic formula has four parameters: Two end points (P_0 at $x = 0$ and P_3 at $x = 1$) and two concavity points (P_1 and P_2). The Bézier polynomial is given by the following expression:

$$B(t) = (1-x)^3 P_0 + 3(1-x)^2 x P_1 + 3(1-x)x^2 P_2 + x^3 P_3, \quad x \in [0, 1]. \tag{7.1}$$

We use this formula as a point-to-point operation between every pair of adjacent extrema. In doing so, N extrema samples are used to create N-1 segments that approximate the complete signal. For each N-1 segment, Equation (7.1) must be applied to the voltages (V) and times (T) of the samples separately. Figure 7.9a illustrates how Equation (7.1) is applied to create a smooth interpolation between extrema values, and, in this particular example, the waveform goes from a local maximum to a local minimum. We let P_0 and P_3 be represented by the voltage–time pairs of the two extrema locations, given by P_0=(V_{max},T_{max}) and P_3=(V_{min},T_{min}).

Concavity points P_1 and P_2 define the smooth transition between the endpoints. They maintain the corresponding voltage values of the endpoints (i.e., V_{max} and V_{min}) to ensure that the derivative at the local maxima/minima is zero, but their time values are set exactly halfway between them in time at T_{ave}=(T_{max} + T_{min})/2. Accordingly, Equation (7.1) is modified to two equations for the voltage and time expressions.

$$V(x) = (1-x)^3 V_{max} + 3(1-x)^2 x V_{max} + 3(1-x)x^2 V_{min} + x^3 V_{min}, \tag{7.2}$$

$$T(x) = (1-x)^3 T_{max} + \frac{3}{2}(1-x)^2 x(T_{max} - T_{min}) + \frac{3}{2}(1-x)x^2(T_{max} - T_{min}) + x^3 T_{min}. \tag{7.3}$$

These expressions create vectors of voltage and time values, providing a smooth curve that is nearly sinusoidal in nature between the two extrema locations.

This process is demonstrated in Figure 7.9b, which shows a composite Bézier-reconstructed signal created in a piecewise fashion following the pattern of a maximum to a minimum to a maximum to the remaining extrema. We have found that the utilization of the Bézier equation is an apt method for reconstructing a signal using local extrema values, and what we have shown here is the simplest application of it. In Section 7.5, we demonstrate that this technique can reconstruct complex waveforms with very small mean-squared error between the original waveform and the reconstructed version.

7.4.2 RAMP Triggering Implementation

Using the reconfigurability available from the RAMP system, we are able to synthesize a circuit capable of locating and sampling extrema values. Figure 7.10 shows the circuit we synthesized for finding local maxima; the local minima detector circuit is a symmetrically equivalent circuit (not shown). Figure 7.11 shows the measured output of the maxima and minima detectors that were synthesized on the RAMP, along with timing diagrams showing the outputs of the various stages of these circuits.

Operation of the maxima detector circuit is as follows. Locating the maximum extrema of the signal is performed by first taking the envelope of the signal. This envelope is set to track the input aggressively on the rising edges and to lag behind when the signal begins to decline. Accordingly, the transconductance $G_{m,A}$ is set to be larger than $G_{m,B}$ to achieve this asymmetry in tracking the input signal that results in envelope detection. The comparator then produces a logic high signal when the envelope lags behind the input—signaling a local maximum. This logic high value is used to trigger a pulse generator which provides the 'event pulse' signal shown in the bottom plot of the Figure 7.11.

An event pulse is triggered to logic high at every extrema occurrence and corresponds to the start pulse signals in Figures 7.1 and 7.3. This signal commences the digitization process in the ADC and TDC and also activates the sample-and-hold to sample the value of the

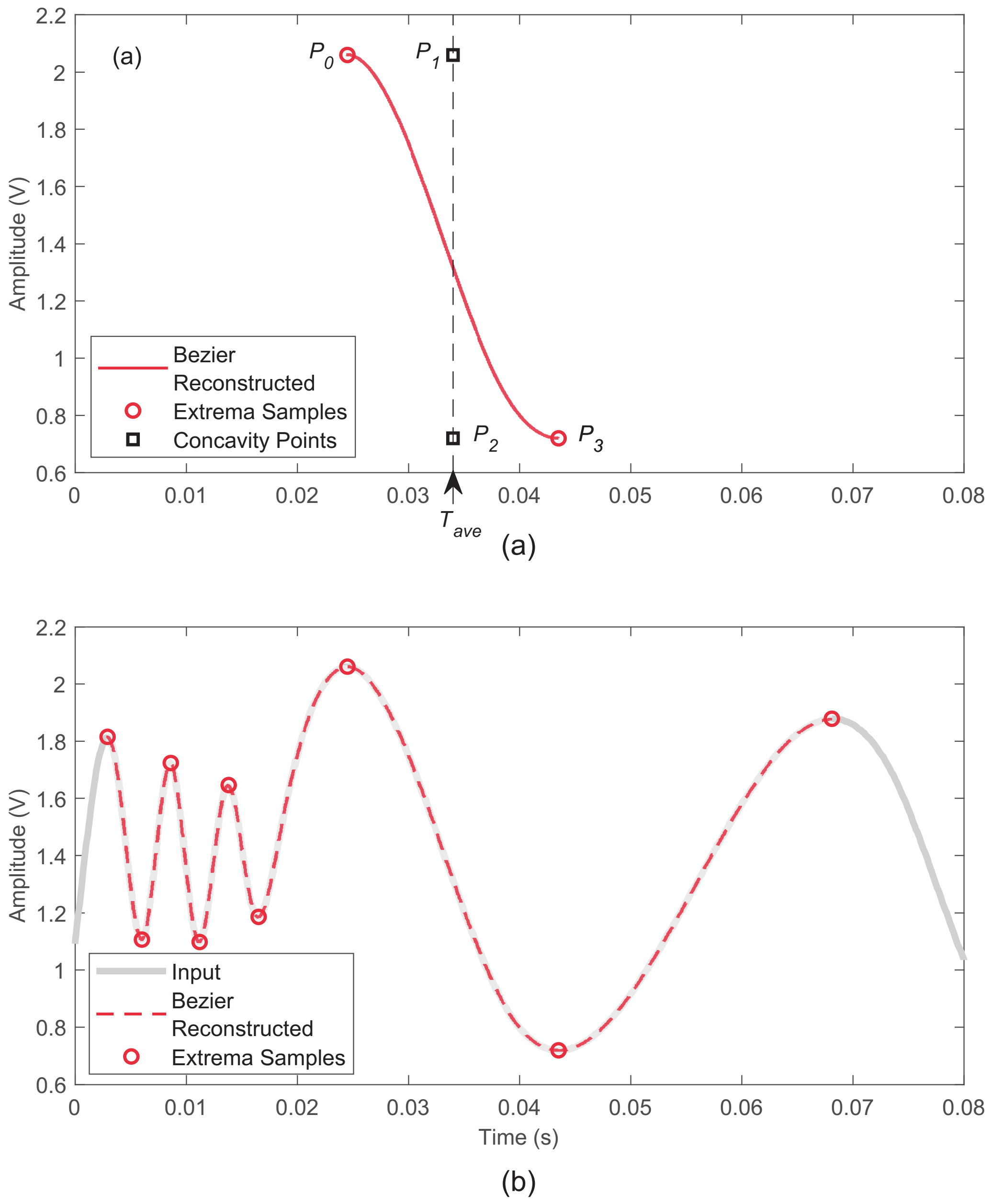


Figure 7.9: (**a**) A Bézier interpolated segment constructed via endpoints P_0 and P_3 with concavity points P_1 and P_2 defining the directional path. T_{ave} is the center point between endpoints P_0 and P_3. (**b**) A continuous piecewise Bézier reconstruction that was generated from local maxima/minima and overlaid on the input signal.

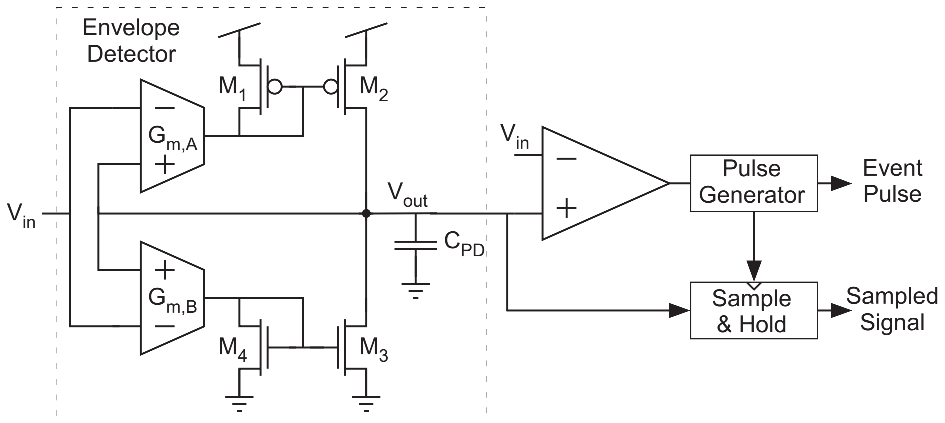


Figure 7.10: The maxima detector circuit, which makes up one half of the analog front-end synthesized in the RAMP system. The minima detector circuit is the symmetric equivalent.

signal. The ADC takes the sampled value from the sample-and-hold circuit and converts the voltage into its corresponding digital codeword. The TDC utilizes the event pulse to halt its operation, allowing readout of the digital codeword representing the time duration since the previous extrema occurrence. The TDC is then reset, allowing it to begin counting up until the next extrema occurrence.

7.5 System Implementation and Example Applications

In this Section, we present the complete fabricated system and its application to three biorelated application examples including voice, electromyography (EMG), and electrocardiogram (ECG).

The designs presented for these applications were measured from fabricated chips. The RAMP was fabricated in a standard 0.35 μm CMOS process, while the ADC/TDC was fabricated in a 0.5 μm CMOS process. Both die photographs are shown in Figure 7.12. Future implementations would combine both the RAMP and the asynchronous ADC/TDC on the same die to improve overall performance. Additionally, scaling the ADC down to a newer technology node would also significantly improve power consumption, particularly since the SA-ADC and TDC are mainly comprised of digital circuits.

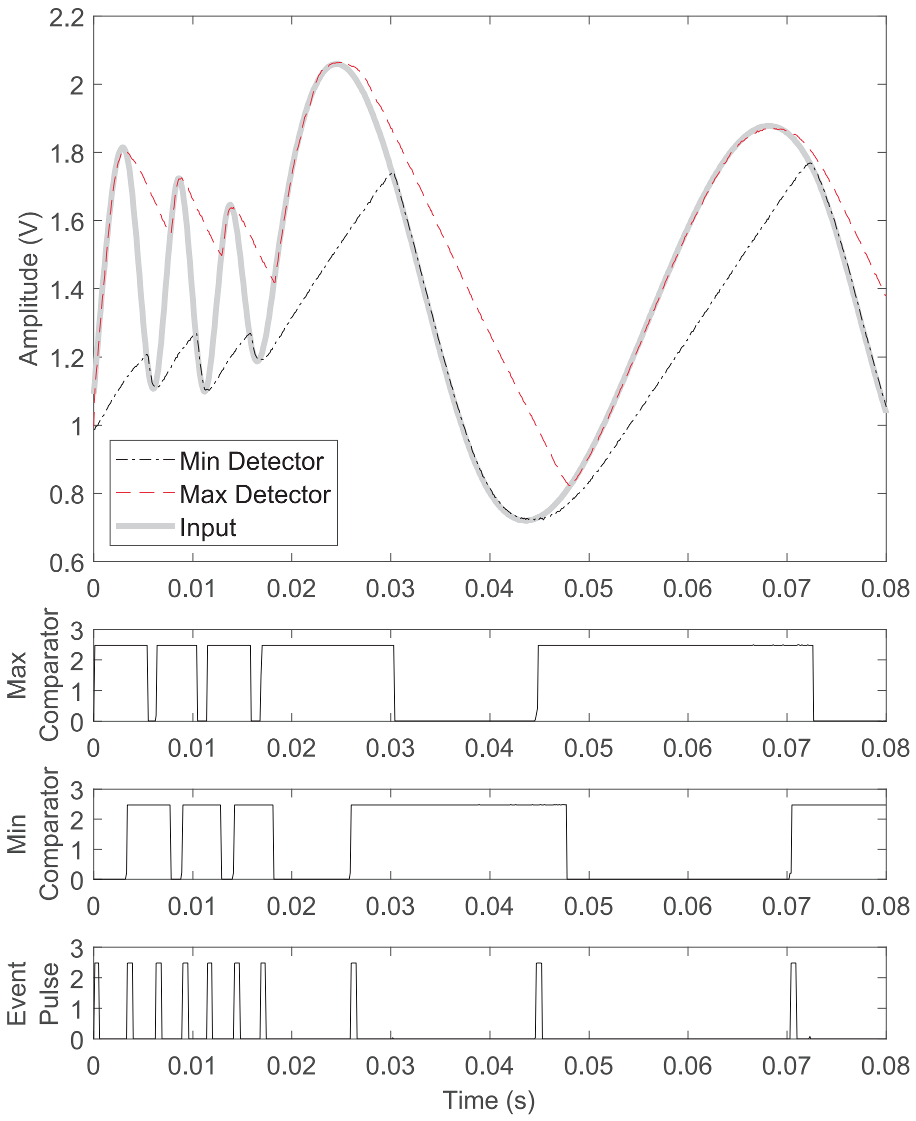
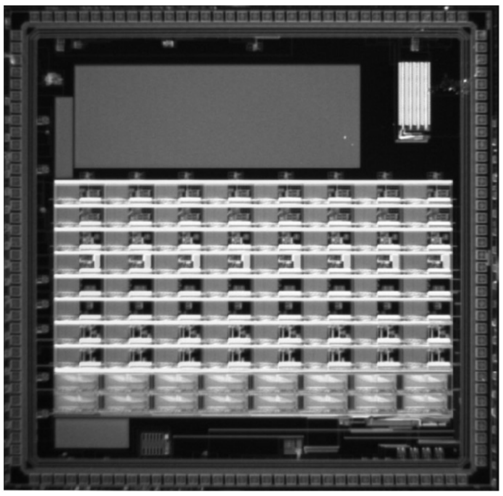
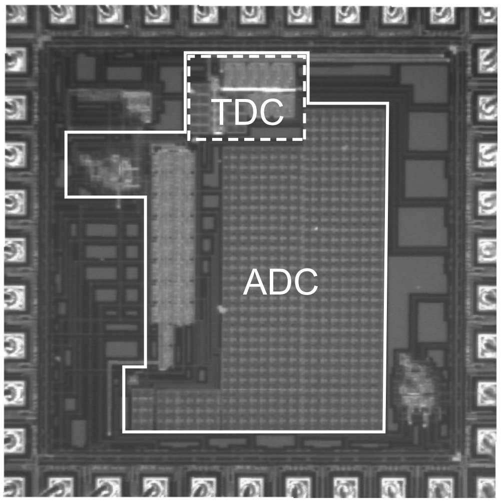


Figure 7.11: The maxima and minima locator circuits find local extrema by comparing the output of an envelope detector with the input signal. The measured timing diagrams indicate when an event (local maximum/minimum) has been detected.


(a)



(b)

Figure 7.12: (**a**) Die photograph of the reconfigurable analog/mixed-signal platform (RAMP) fabricated in a 0.35 μm standard CMOS process. (**b**) Die photograph of the asynchronous ADC/TDC fabricated in a 0.5 μm standard CMOS process.

7.5.1 Power Consumption

In general, the RAMP power consumption is dependent upon the circuit that is synthesized for the front-end application. In the examples of this Section, the signals can be characterized as biorelated signals with low frequency components (i.e., $f < 10k$Hz). By performing this extrema location preprocessing in the analog domain for these biorelated signals, we were able to detect extrema at a measured static power consumption of 4.95 μW [8].

The ADC has an extremely low static current draw, measured with an overall static power consumption of 14.75 nW. The low static power consumption means that the ADC can remain powered on between samples without worry of draining the power budget. Energy per conversion was too low to be measured experimentally, so the energy per conversion was obtained from simulation as 47.4 nJ.

The TDC can operate with frequencies ranging up to tens of kilohertz, but, for the example applications shown here, the TDC was operated at a frequency of 1.15 kHz with a measured power consumption of 1.01 μW. It should be noted that this component will greatly benefit from smaller CMOS technology nodes and lower power supplies.

A summary of the performance of the fabricated RAMP, SA-ADC, and TDC is provided in Table 7.1.

Table 7.1: Summary of ADC and TDC specifications.

Extrema Sampling Ramp Implementation	4.95 μW
SAR ADC Static Power Consumption	14.75 nW
SAR ADC Energy/Conversion	47.4 nJ
SAR ADC Resolution	10 bits
TDC Power Consumption	1.01 μW @ 1.15 kHz
TDC Resolution	10 bits

7.5.2 Voice Recording

The first application demonstrated is the capture and quantization of extrema values within a speech recording. Low-power speech sampling and reconstruction is useful in IoT devices which actively listen for predefined commands. In these applications, perfect reconstruction is not necessary. Instead, only a certain mean-squared error threshold must be respected so the system can identify known commands.

Figure 7.13 shows the successful capture of 1436 extrema values in a 1.45 s speech waveform. The fastest captured component of the signal is 5333 Hz, thus dictating a minimum of 15,528 samples to be taken in a traditional Nyquist-rate sampling system. Extrema sampling results in a nearly eleven-times reduction in sample values and maintains a 0.0483 mean-squared error rate when reconstructed using the Bzier interpolation technique of Section 7.4.

The reduced number of samples provided by the asynchronous sampling can significantly reduce the overall power consumption of the system, both at the conversion stage and at the subsequent digital processing stage when the data are analyzed. To provide a comparison in the power consumption of only the data-conversion stage, we can use the power and energy values provided in Table 7.1 to determine the power savings of the asynchronous ADC over an identical ADC continuously converting all data at a fixed rate. The average power consumption of the data converter is given by

$$P = P_{RAMP} + P_{ADC} + P_{TDC} + \frac{nE_{conv}}{t}, \tag{7.4}$$

where the P values are the static power consumption of the associated stages (with $P_{RAMP} = 0$ for a fixed-rate converter), n is the number of samples taken over a time interval (t),

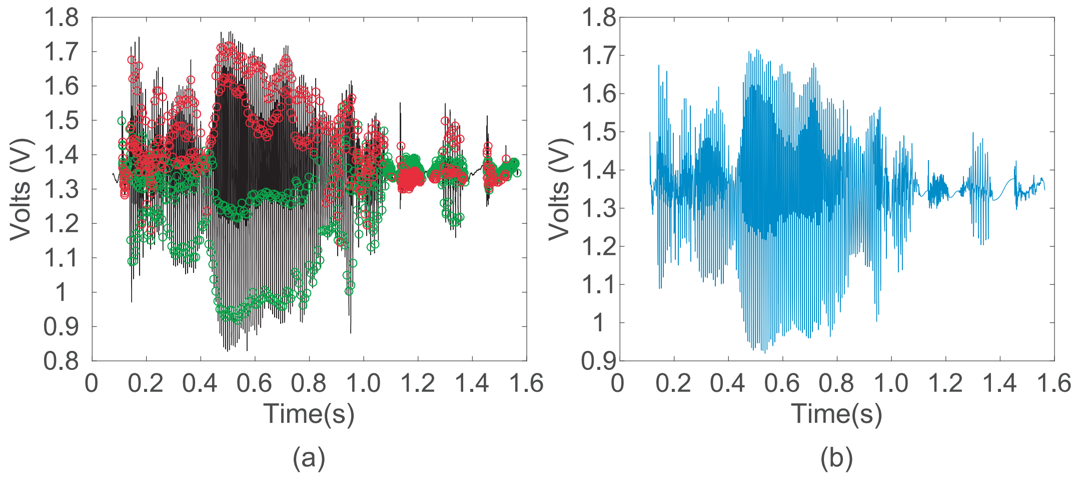


Figure 7.13: (**a**) The original vocal waveform with successfully detected peaks. (**b**) The recreated vocal waveform using Bzier reconstruction.

and E_{conv} is the energy required by the ADC to convert one sample. As a simplification, we let P_{TDC} also represent the power consumed by a fixed-rate clock for a traditional ADC. Note, however, that a fixed-rate ADC would need a faster clock than the asynchronous case, thereby meaning that the fixed-rate ADC would likely consume even more power.

Based upon the values in Table 7.1 and Equation (7.4), our asynchronous converter would consume an average of 52.9 μW, while a fixed-rate ADC (without any analog preprocessing) would consume an average of 509 μW for this voice-recording application. As a result, the asynchronous ADC consumes 9.6-times less power than a fixed-rate converter. While these specific numbers depend on the exact characteristics of the ADC that is used, the fundamental principle holds that reducing the number of samples that are converted can significantly decrease the overall power consumed by an ADC, as is clearly seen by the last term in Equation (7.4). As a result, using even-more power-efficient ADCs along with analog preprocessing would result in even further savings. Furthermore, in this voice-recording example, the speech is continuous throughout the 1.45 s waveform. For many real-world applications, pauses in the speech would be present, which would further reduce the power consumed by the asynchronous ADC, but not the fixed-rate ADC.

7.5.3 Electromyography

The next application involves the quantization of an electromyography (EMG) signal. EMG signals are traditionally used to diagnose muscle and nerve health. In addition to

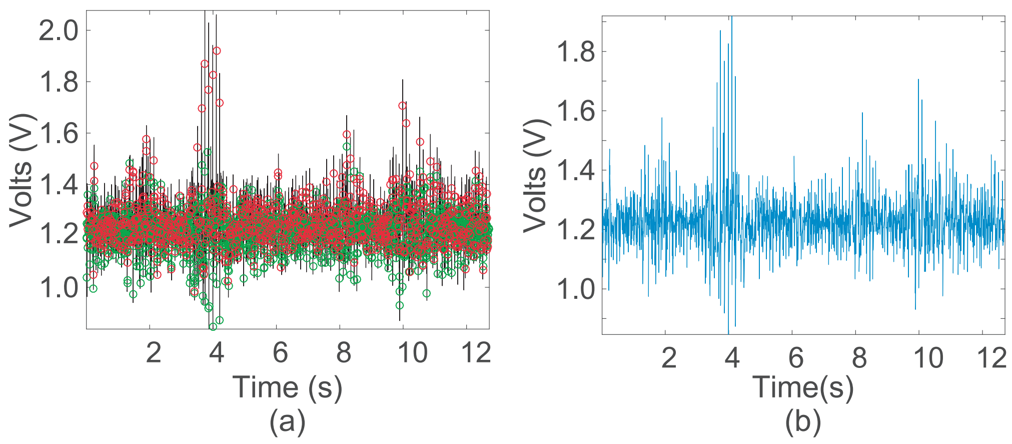


Figure 7.14: (**a**) The original electromyography (EMG) waveform with successfully detected peaks. (**b**) The recreated EMG waveform using Bzier reconstruction.

measuring neuro-muscular health, EMGs are gaining popularity in measuring muscular exertion in physical therapy and sport-science applications [78]. As these applications become more advanced, increasingly power- and sample-efficient devices will be required.

Figure 7.14 demonstrates the results of our presented system on an EMG waveform taken from the example EMG database found at [79]. Over the 12.7 s signal, 3035 extrema values were successfully detected. Given that the highest frequency within the sample was approximately 2 kHz, then 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-squared error in reconstruction (using Bzier reconstruction) of less than 0.0036. Applying Equation (7.4), the asynchronous ADC has a factor of 11-times less average power consumed than the fixed-rate ADC for this EMG example.

7.5.4 Electrocardiogram

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 that 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 [80]. For the purpose of demonstrating this system, an extrema sampling approach

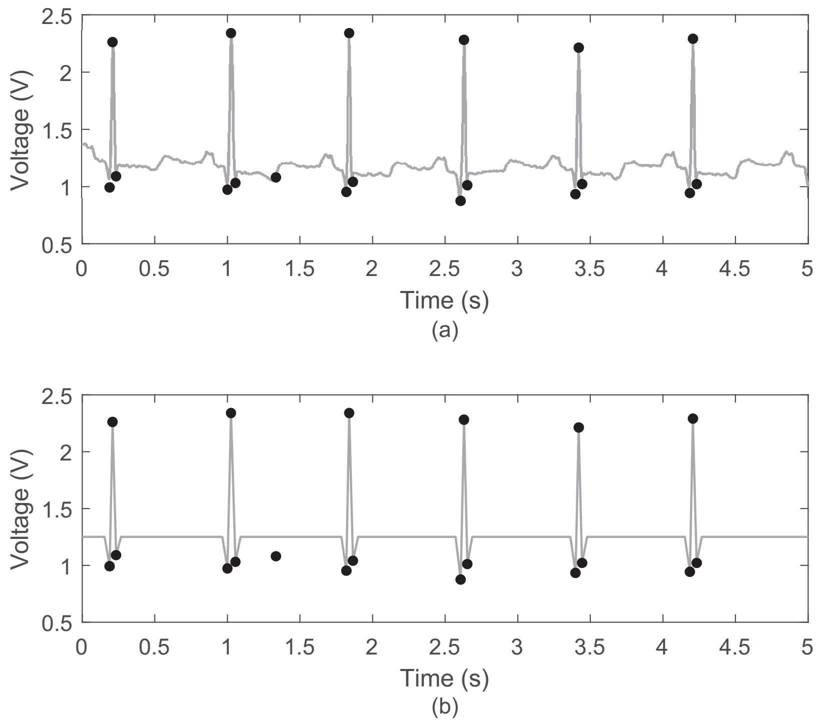


Figure 7.15: (**a**) The original ECG waveform taken from the MIT arrhythmia database with the detected extrema values shown as dots. (**b**) QRS-complex reconstruction using a simplistic quasi-linear model based on QRS waveform shapes (line) with the detected QRS values (dots) as well as a single false positive.

is a natural fit for reducing an ECG waveform to its QRS-complex—which can be viewed as 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 7.15 shows the operation of the system and the resultant approximations of the amplitudes and time intervals between the QRS peaks. The QRS waveform of Figure 7.15b was not reconstructed using Bzier reconstruction—a simplistic interpolation technique based upon QRS shapes is adequate to glean the medically-relevant information from the QRS-complex for this particular case. The device was tested with real world data taken from the MIT arrhythmia database [79].

Figure 7.15 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 57 Hz change would necessitate a sampling rate of 114 Hz from a conventional Nyquist-rate ADC.

This sampling rate would yield approximately 485 samples over the 4.228 s that the signal is demonstrated over. That amount of superfluous samples is greater than 25 times the number of samples taken in our system, even including the single false-positive. Since the frequency of sampling is much lower than for the voice and EMG cases, the amount of power savings using Equation (7.4) is not nearly as significant. However, two points should be noted. First, most ADCs that are designed for heart-rate monitoring use faster sampling rates than the absolute minimum from this example. For example, [81, 82, 83, 84] use sampling rates from 300 Hz to 600 kHz, and using these larger sampling rates would significantly increase the average power consumption in an equivalent ADC. Second, the purpose of many heart-monitoring systems is to determine the features of the QRS complex—while our asynchronous ADC has minimal power savings strictly at the conversion stage for this application, our asynchronous ADC is performing many of the operations that would be reserved for a subsequent digital processing system when using a traditional ADC, which would also have a substantial power consumption. Some application-specific QRS detector systems have been shown to acquire and process heart-rate data at very low power; for example, the system of [84] is able to operate at 220 nW under a 300 mV power supply. However, such a system lacks the flexibility afforded by our reconfigurable analog preprocessing. Additionally, scaling both the technology node and power supply of our ADC (to levels similar to [84]) would also help to significantly further reduce the overall power consumption.

7.6 Conclusions

To reduce the power consumption of an analog-to-digital conversion system, we have introduced an analog preprocessing stage prior to an asynchronous converter to find the data values that are most important to be converted for a given application. Since the timing of the data values is not known a priori, the analog preprocessing triggers the asynchronous converter to acquire a sample, and a time-to-digital converter measures the intersample time period. By converting only the data values that are needed, a significant amount of power can be saved by the system, both at the conversion stage, as well as further down the signal-processing and/or transmission chain. We also demonstrated a sampling technique based upon finding local maxima/minima, and a variety of other triggering operations are also possible due to the reconfigurable nature of the analog front-end.

Chapter 8

Reconfigurable Analog Mixed-Signal Processor

This chapter will be wholly dedicated to the RAMP. Many of its applications have touched the past chapters of this work including the high-side load switch of Chapter 3, FG temperature compensation of Chapter 4, and asynchronous front-end for analog-to-digital conversion in Chapter 7. The contents presented in this chapter were not germane to previous topics, but justify its own chapter. It will include miscellaneous application circuitry, hardware and software infrastructure, and enhancements from past versions of FPAAs developed in this research lab.

8.1 Software Infrastructure

The software infrastructure is critical for facilitating usability for new users of the RAMP – whether it be for undergraduate students in an electronics laboratory or seasoned analog designers in industry. The software toolchain can largely be divided into two categories: front-end and back-end. The front-end is the human-interfacing aspect of the software that codifies the user-defined design in code and passes the information to the back-end application. The back-end realizes the hardware design on the RAMP from the user-defined source code, and additionally is used to establish a means for a communication link between the RAMP and the computer.

A large aspect of the front-end design consists of a simplistic netlist-based language. This design was chosen since it fits nicely within the context of signal-flow design wherein

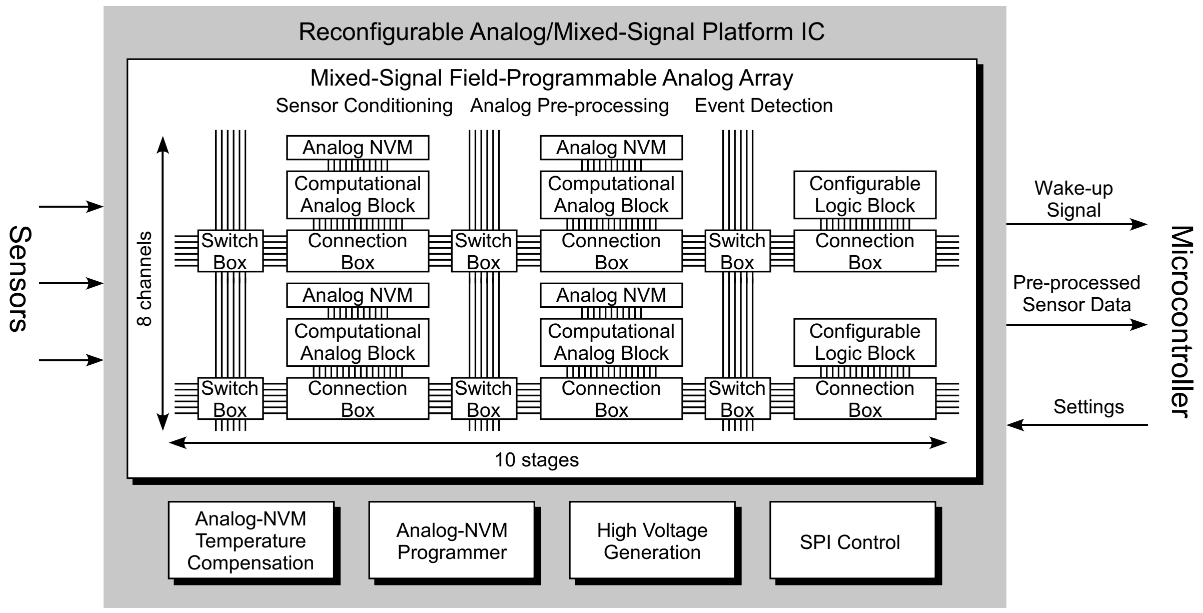


Figure 8.1: Architecture of the reconfigurable analog/mixed-signal platform (RAMP) integrated circuit.

the user defines in code the various circuit elements for the signal to 'flow' through before reaching its output. As shown in Figure 8.1, the signal can pass through one or more of the 80 computation analog blocks (CABs) that are interconnected via ten stages in eight channels, and is organized by the processing type — transconductors, band-pass filters, data conversion, etc. This provides flexibility and parallel processing that can then be fed to a microprocessor further down the signal chain. Moreover, many of the synthesizable elements are tunable to set desired performance parameters and can be specified within the netlisting language.

The netlisting language enumerates the electrical connectivity among the circuit elements of a design. An example of the netlisting language is shown below in 8.1, and also happens to be the source code for generating the reconfigurable analog-to-digital preprocessor signals of Figure 7.11 — placed below for ease of viewing. The two demonstrate the aforementioned signal-flow design.

Listing 8.1: Example netlisting code

```
%==============Input Definition=================
% Define the the input signal pin (A0) and     |
% declare an alias for the signal name (InSig)|
%==========================================
A0 InSig

%========PeakDetector Parameter List=========
% PeakDetector In Out Chan Stage Atk Dec    |
%==========================================
PeakDetector InSig PkEnvMax 0 1 20e 9 .30e 9
PeakDetector InSig PkEnvMin 3 1 .35e 9 5e 9

%========Comparator Parameter List===========
%     Comparator Plus Neg Out Chan Bias     |
%==========================================
Comparator PkEnvMax InSig MaxFound 0 250e 9
Comparator InSig PkEnvMin MinFound 3 500e 9

%======Pulse Generator Parameter List========
%    PulseGenerator In Out Bias Time Chan   |
%==========================================
PulseGenerator MaxFound MaxPulse 1000e 9 .6e 9 1
PulseGenerator MinFound MinPulse 100e 9 .35e 9 3


%===========Combinational Logic==============
% Map input and output signals for digital  |
% logic gates within Look Up Table          |
%==========================================
LUTx_S8C2 MaxPulse MinPulse Gnd Gnd Gnd Gnd ConvNow Gnd
LUT_S8C2 Y0=X0|X1

%============Output Definition===============
% Map output signal (Y0) is output pin (ADC0) |
%==========================================
```

ADC0 Y0

Beginning with the input, we must define which pin to receive the input signal. In the case of 8.1, pin A0 was chosen receive the sinusoidal input shown in the top plot of Fig. 8.2. This input will flow into two separate Peak Detector components sharing the same stage 0, but splitting into two channels 0 and 3, respectively. Each of their respective component parameters are enumerated after the component name including: input signal alias, output signal alias, channel number, stage number, followed finally by attack and decay currents. The attack and decay currents are tuneable via floating-gate transistors. Of note is the complementary attack and decay current parameters of the respective the min and max peak detectors that contribute to their functionality.

The signal-flow design of this example continues through a set of comparators, pulse generators, and combinational logic gates, where it terminates its output at pin ADC0. As the pin name suggests, it can be wired a microcontroller ADC. While the code example of 8.1 demonstrates the relationship of electrical connectivity among the components, it also demonstrates macro-level abstraction of the netlisting language that promotes code re-use. Each one-line macro represents a component must be fully defined in the netlist document for the compiler. For example, the macro definition for the comparator is as follows.

Listing 8.2: Example netlisting Modular Macro code

```
%=====Comparator Macro Definition=======
begin Comparator Plus Neg Out Chan Bias
    Cmp2_S6C<Chan> <Plus> <Neg> <Out>
    FG_S6C<Chan> Cmp2_Bia Itar=<Bias>
end
```

The macro is composed of primitive components that can be interpreted by the compiler and translated to bytecode for RAMP programming. The designer should be cognizant of the overall design when choosing from available components to reduce loading effects on the circuit. Synthesizing components among a number of different stages/channels is unavoidable due to the nature of the RAMP, but there are component choices that best optimize the design or should dictate the priority because their performance is the most sensitive to loading effects.

In the netlist example of 8.1, the peak detector outputs feed into the same respective

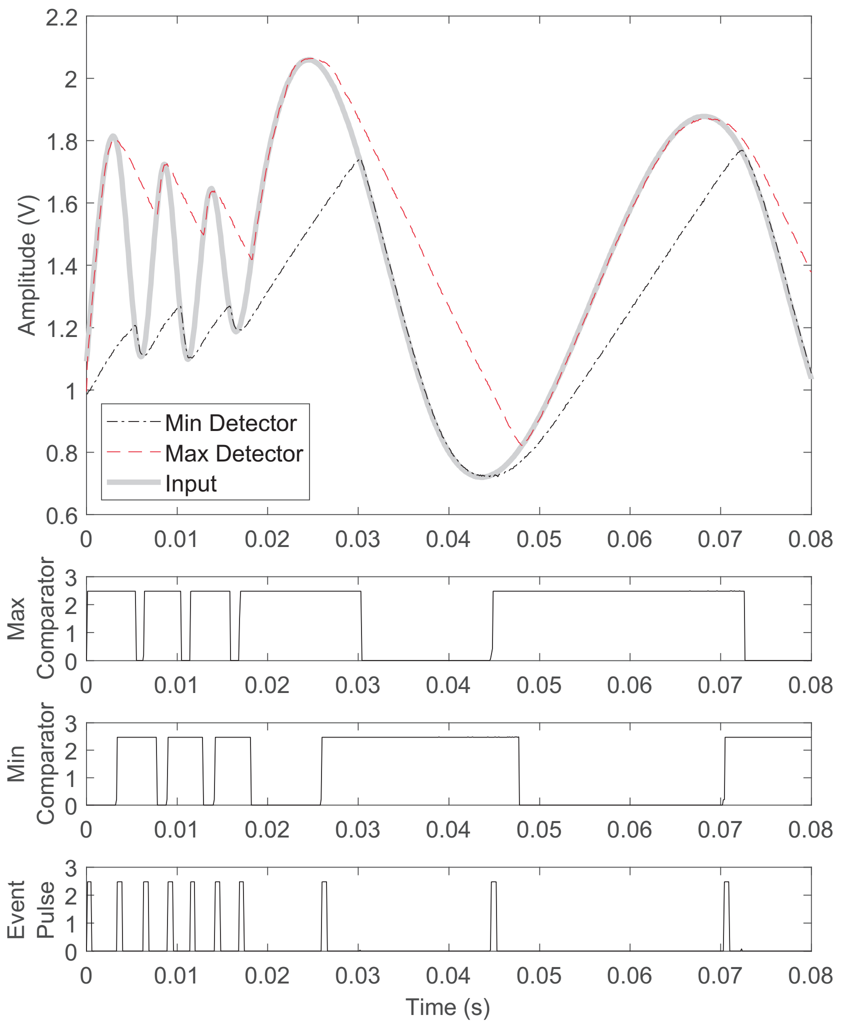


Figure 8.2: The maxima and minima locator circuits find local extrema by comparing the output of an envelope detector with the input signal. Each output signal shown corresponds to an output of a synthesized RAMP component (e.g. pulse generator) that flows into another component or the final output.

channels of their paired comparators. However, note that the comparator macros do not explicitly define their stages as an input parameter. Instead, they are defined as a primitive component in their macro definition shown in 8.2. The signals are now being processed in stage 6, which is more than halfway through the chips towards its terminating output pin.

All associated back-end software design — sometimes referred to as physical design flow — is compatible with Matlab, and it's free and open-source alternative, Octave. The back-end software emphasizes processes that renders a synthesized electronic design on the RAMP without human intervention. Without the ability to batch process the rendering of the netlist to a ready-for-deployment state greatly reduces the RAMP's utility in a number of ways. The first, is that this process would be too tedious and error-prone for a human to arrive at an optimum placing and routing signal-flow design — each of which have their own set of challenges. Furthermore, back-end software greatly reduces the barriers to in-the-field programming. Both attribute to the end-user to focus solely on the electronic design and to rapidly prototype it in silicon.

The back-end software accepts the user's netlist file as an input parameter and checks it against a set of syntax rules. Once this check has been passed inspection, the design may be passed to a place routine whose output is then passed to a route routine. A place routine optimizes the placement of the user-specified component list within the constraints of available components within the active chip area. Its objective is to arrive at a placement solution that will permit the best electronic performance and allow for a viable routing solution. The wire routing which is handled by the routing procedure iteratively optimizes the physical connections among the placed components as defined in the netlist. Both procedures utilize heuristic algorithms to arrive at their optimum solutions.

8.2 RAMP Version 1.1 Enhancements

As the version name suggests, RAMP version 1.1 encompasses iterative enhancements as opposed to a complete overhaul of predecessor's version. The largest architectural change to the latest version is the inclusion of an eleventh stage referred to as the 'Unique Stage'. It's a departure from the original stages in that each CAB contains a different application-specific circuit. This exists all while maintaining a backwards-compatible infrastructure that preserves the ability to interface and program the CAB elements.

The philosophy of the original RAMP v.1.0 design was the ability to process signals in

Channel	Circuit
0	Temperature Compensation
1	Subthreshold Neurons
2	BiQuad Filter
3	Automatic Gain Control
4	Bandgap Reference with Temperature Compensation
5	Wheatstone Bridge
6	Ring Oscillator
7	Neural Amplifier

a re-configurable, parallel manner. The parallel-aspect is realized by having 8 channels of repeated CABs for each stage. Much like the example of netlist 8.1, this architecture-choice allowed us to design two parallel, but complementary signals to determine maximum and minimum amplitudes of a voltage signal. However, there are a number of cases in which synthesized circuitry has no benefit from parallelization and should be a standalone element.

One reason is that there are common infrastructural elements that do not benefit the overall design from a piecemealed synthesis. For example, it could potentially take away resources greatly needed by the design like routing options and be consuming other valuable circuit elements. A second is some circuits rarely need/utilize a duplicate of itself. Having multiples of the same CAB elements along each channel would be wasteful use of silicon space. And finally, the unique stage allows for evaluating contemporary circuit alternatives and exploratory elements. This would allow an evaluation of said CAB elements within the same infrastructure for performance comparisons. For these reasons became the motivation for a unique, eleventh stage in the RAMP v.1.1 whose CAB elements are enumerated below by channel.

The temperature compensation CAB provides the temperature compensation for all programmable current sources and is fully elaborated upon in Chapter 4. The neuron circuitry contains two complete neuron circuits. These circuits have subthreshold, current-mode operation like the ones of [85]. These neurons can leverage our previously developed programming infrastructure which is needed for neural-network development. When neurons are typically rendered in CMOS, they're in the form of nonvolatile analog (via current) memory, but need the ability to "learn." Each neuron's memory allows one to improve the feedback network's performance in solving a problem which is done by re-programming the neuron (i.e. floating-gate transistor). These neurons become the basis for future neural-network development on the RAMP. Similarly, CAB element of channel 7 consists of a neural amplifier which will permit real-time neural recording activity. The amplifier is specifically designed to meet the needs of this application-specific requirements.

The BiQuad Filter is a programmable band-pass filter like ones employed in the filtering stage which utilize C4 filters. The the unique stage BiQuad filter fulfills the third aforementioned rationale in evaluating their performance within our current infrastructure. Similarly, channel 4 CAB element consists of a bandgap reference with temperature compensation and will allow us to evaluate its performance against our current bandgap reference.

The ring oscillator is a common circuit used as a clock source. The one of channel 6 is current-controlled by a floating-gate current source which allows re-programmability and low-power consumption. The inclusion of this circuit in the unique stage met the criterion for the first rationale.

The last two elements of the unique stage are common sensor-interfacing circuitry with unique input signaling requirements, but whose post-processing could benefit from the collection of circuits offered by the RAMP's ten previous stages. For example, the wheatstone bridge circuit requires a two-terminal resistive element input, which does not fit the typical single-ended voltage inputs of the RAMP. The automatic gain control circuit of channel 3 was designed for a MEMs microphone input. It's transfer function characteristics include a small-signal amplification and large-signal compression due to the inherent nature of microphone usage. It's design consists eleven programmable biases, four OTAs, and two opamps. If it were to be synthesized in the RAMP rather than its own CAB element, it would have occupied a lot CAB elements and routing lines. While not necessarily efficient, it would take a lot of planning to implement additional processing circuits within the synthesized design. Overall auditory processing remains a large application of the RAMP.

8.3 Conclusion

The RAMP is unique, rapid-prototyping device that can provide re-programmable, in-silicon circuitry for front-ends and sensor interfacing. It is more generally categorized as a Field-Programmable Analog Array that additionally provides common digital and mixed-signal circuitry. There are two versions of the RAMP, where the successor offers incremental improvements over the original in the form of improved circuit selection, infrastructure, and the inclusion of experimental circuitry for future iterations.

The software toolchain permits end-users to rapidly prototype designs with a process very closely follows the typical FPGA design synthesis. The software permits a low barrier-to-entry with software interfaces such as well-known Matlab programming along with its free counterpart, Octave. Furthermore, the place-and-route procedures allow a designer to concentrate on the electronic circuit design from a schematic point-of-view allowing for more usability among novice designers without the physical knowledge of electronic design.

Chapter 9

Conclusions and Future Work

The push for big data and sensing the world around us will be realized by networking traditionally offline and bringing resource-constrained devices online. The viability of these devices will depend on extremely power-efficient devices whose battery dependent designs have a lifespan of months and years as opposed to weeks. This constraint is not unrealistic for environmental-sensing that could be placed in difficult-to-reach locations.

Prior electronic sensing efficiency gains were largely owed to shrinking feature sizes which greatly benefit the digital domain. The shrinking transistor sizes could operate on lower supply voltages, and for applications like digital signal processing and mixed-signal circuitry, operations became more and more efficient with each generation of downsizing. However, the rates of downsizing into sub-10nm has been limited in recent years, as well as their impact on power savings from lower power supplies. For example, the nominal supply voltage drop from a typical 65nm node (first released circa 2005) to a 7nm node (first released by TSMC circa 2018) was about 500mV to 0.75V [86].

An alternative low-power sensing strategy presented in this work is to keep much of the desired signal's computation in its native domain. Therefore, when sensing the natural world, the signals should processed in the analog domain for as long as possible to achieve low-power computation as opposed to immediately digitizing the signal at the front end. Many of the hindrances with analog signal processing is that the underlying IC designs are application-specific and not flexible in their post-fabrication form. These are two contributing factors for why most designs are conversely kept in the digital domain in the form of microcontrollers and FPGAs.

This work's contributions are towards lowering the barrier to entry for general-purpose and reconfigurable analog signal processing in the form of FPAAs. Because FPAAs are system-level chips, there were many infrastructural elements in need of improvement or were completely lacking. Much of the infrastructural developments was in service to analog memories employed as programmable biases. Floating-gates for analog applications have traditionally not been deployed in commercial applications, and as a result, is not well-understood among IC designers and is under-represented in literature. Furthermore, larger challenges are presented when there are a large arrays of analog floating-gates.

One of the first challenges we undertook towards the objective of having an in-the-field-capable FPAA was a high-side load switch for FG programming. The two supply voltages represent the two modes of operation for floating-gates: programming and run-mode. The high-side switch allowed seamless switching between the two even though the programming voltage is almost three times larger than the run-mode voltage. This on-chip device obviated the need for external bench-top equipment to carry out FG programming, and as result allows allows for in-the-field changes.

Another contemporary issue with FG current sources addressed in this work is temperature compensation. When an FPAA is placed in-the-field, it will likely not experience a constant room temperature. Unfortunately, an operational FG current sources have an exponential dependence their ambient temperature. This work presented a temperature compensation scheme for FPAA-like environment that will likely have multiple and different current biases for each design. Furthermore, the temperature compensation scheme was characterized such that the most optimized conditions for each design could be chosen with a low design overhead.

Further improvements for low-overhead and accurate FG programming was detailed in this dissertation. While various methodologies of FG programming have been published, there had not been an in-depth discussion in comparing results among various different works. Specifically for the sake of future FPAA designs like the RAMP, there was a need for a methodology to compare above-ground and nascent below-ground programming. Below-ground programming accuracy and analysis had not been published before this work.

Combining all the infrastructural elements together on top of prior work done to the RAMP, we have a fully working reconfigurable analog system. In demonstrating the full system and the wake-on event processing, the final work presented an asynchronous ADC sampling paradigm which leverages the advantages of analog signal processing for power savings in data conversion.

9.1 Future Work

There are areas in this work that are unexplored and elements that need to be taken further to bring their promise to complete fruition. The work that deserves the largest attention is the development of a system-level below-ground programmable RAMP. The work done in Chapter 6 successfully demonstrated its viability as a reconfigurable platform. But this proof-of-concept only had 16 FGs that biased C4 filters. However, the infrastructure used in the proof-of-concept was designed in such a way that it could be successfully scaled up to the system-level without any issues. An additional improvement that was explored in Chapter 6 and could be implemented in the next iteration RAMP was the single, high-gain feedback programmer. Integrating this design will provide more consistent programming results among all FGs in the array.

Probably the larger challenge for developing a below-ground programmable RAMP is a tedious task of redesigning it for a smaller process node. The RAMP 2.0 that was discussed in Chapter 8 was developed in a 0.35μm technology, which is increasingly being taken out of production at the time of this writing. And most importantly, FG characterization needs to take place to determine proper programming procedures and operation before a RAMP design can be sent out to fabrication. Recall that FGs are not a typical circuit element supplied in a process design kit. While this is a mild hindrance, it is undoubtedly a large time investment.

Chapter 4's temperature compensation performance analyzed how diverse biasing led to a larger percent error in output current performance. A refinement of the temperature compensation scheme could be expanded to multiple temperature compensation reference lines. This is a viable approach because of the low overhead in the temperature compensation scheme design. Moreover, additional schemes could be explored and developed like a constant G_m compensation. And finally, the largest omission in this work is the lack of a temperature-independent current-source, where the original design utilized an external source to demonstrate the proof-of-concept.

Finally, the RAMP is a mixed-signal platform that could have even more opportunities to off-load MCU computation and consequently keep the MCU in a sleep state longer if it had some digital processing capabilities. The first step towards this goal would be to integrate an ADC like the one presented in Chapter 7 which has the ability to operate with periodic sampling and asynchronous sampling. The second step would be to integrate an ALU and control logic to handle integer math. Some basic instructions could give the end designer even more possibilities with a more balanced analog and digital interface.

References

[1] M. M. Navidi, “Integrated circuits for programming flash memories in portable applications,” Ph.D. dissertation, West Virginia University, 2018.

[2] M. M. Navidi and D. W. Graham, “A regulated charge pump for injecting floating-gate transistors,” in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2017, pp. 1–4.

[3] J. Chang, Y. Chen, W. Chan, S. P. Singh, H. Cheng, H. Fujiwara, J. Lin, K. Lin, J. Hung, R. Lee, H. Liao, J. Liaw, Q. Li, C. Lin, M. Chiang, and S. Wu, “12.1 a 7nm 256mb sram in high-k metal-gate finfet technology with write-assist circuitry for low-vminapplications,” in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, Feb 2017, pp. 206–207.

[4] B. Rumberg, D. W. Graham, S. Clites, B. M. Kelly, M. M. Navidi, A. Dilello, and V. Kulathumani, “Ramp: accelerating wireless sensor hardware design with a reconfigurable analog/mixed-signal platform,” in *Proceedings of the International Conference on Information Processing in Sensor Networks.* ACM, 2015, pp. 47–58.

[5] 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.

[6] 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.

[7] 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*, pp. 1–9, 2016.

[8] B. Rumberg and D. Graham, “A low-power field-programmable analog array for wireless sensing,” in *International Symposium on Quality Electronic Design*, March 2015, pp. 542–546.

[9] *Particle Xenon Datasheet*, Particle, October 2018.

[10] *TelosB Mote Platform*, Memsic, rev. A.

[11] *Mixed-Signal Microcontrollers*, Texas Instruments, 7 2018.

[12] J. Graeme, "Tame transducer bridge errors with op amp feedback control," *EDN*, p. 173176, May 1982.

[13] S. Franco, *Design with Operational Amplifiers and Analog Integrated Circuits*, 4th ed. McGraw-Hill, 2015.

[14] *Low Quiescent Current LDO*, Microchip Technology, 9 2016, rev. D.

[15] *High Side Power Switches*, Micrel, July 2012.

[16] T. Skovmand, *Micropower High Side MOSFET Drivers*, Linear Technology, January 1993.

[17] *Ultra-Low Voltage Energy Harvester and Primary Battery Life Extender*, Linear Technology, December 2013.

[18] R. Gariboldi and F. Pulvirenti, "A 70 milliohm intelligent high side switch with full diagnostics," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 7, pp. 915–923, Jul 1996.

[19] A. Danchiv, M. Hulub, and D. Manta, "An area efficient multi-channel high side switch implementation," in *Proceedings of the ESSCIRC*, Sept 2011, pp. 327–330.

[20] H. Marian, M. Cristian, O. Ionut, S. Mihai, and D. Andrei, "Short circuit protection in dual configurable high side switch," in *International Semiconductor Conference*, vol. 2, Oct 2011, pp. 417–420.

[21] J. Shin, I.-Y. Chung, Y. J. Park, and H. S. Min, "A new charge pump without degradation in threshold voltage due to body effect," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 8, pp. 1227–1230, Aug 2000.

[22] H. Ballan and M. Declercq, *High voltage devices and circuits in standard CMOS technologies.* Springer Science & Business Media, 2013.

[23] P. Hasler, A. Andreou, C. Diorio, B. Minch, and C. Mead, "Impact ionization and hot-electron injection derived consistently from boltzmann transport," *VLSI Design*, vol. 8, no. 1-4, pp. 454–461, 1998.

[24] 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.

[25] 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.

[26] 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.

[27] B. Rumberg and D. W. Graham, "A floating-gate memory cell for continuous-time programming," in *Midwest Symposium on Circuits and Systems.* IEEE, 2012, pp. 214–217.

[28] D. Kahng and S. Sze, "A floating gate and its application to memory devices," *Bell System Technical Journal, The*, vol. 46, no. 6, pp. 1288–1295, July 1967.

[29] E. Sackinger and W. Guggenbahl, "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.

[30] L. Carley, "Trimming analog circuits using floating-gate analog mos memory," *IEEE Journal of Solid-State Circuits*, vol. 24, no. 6, pp. 1569–1575, Dec 1989.

[31] S. Mittal and J. Vetter, "A survey of software techniques for using non-volatile memories for storage and main memory systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. PP, no. 99, pp. 1–1, 2015.

[32] S. Shah, H. Toreyin, J. Hasler, and A. Natarajan, "Temperature sensitivity and compensation on a reconfigurable platform," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 3, pp. 604–607, March 2018.

[33] C. Mead, *Analog VLSI and Neural Systems*, ser. Addison-Wesley VLSI system series. Addison-Wesley, 1989.

[34] M. Holler, S. Tam, H. Castro, and R. Benson, "An electrically trainable artificial neural network (etann) with 10240 'floating gate' synapses," in *International Joint Conference on Neural Networks*, 1989, pp. 191–196 vol.2.

[35] T. Shibata and T. Ohmi, "A functional mos transistor featuring gate-level weighted sum and threshold operations," *IEEE Transactions on Electron Device*, vol. 39, no. 6, pp. 1444–1455, Jun 1992.

[36] B. Minch, C. Diorio, P. Hasler, and C. Mead, "Translinear circuits using subthreshold floating-gate mos transistors," *Analog Integrated Circuits and Signal Processing*, vol. 9, no. 2, pp. 167–179, 1996.

[37] P. Hasler, B. Minch, and C. Diorio, "Adaptive circuits using pfet floating-gate devices," in *Proceedings Conference on Advanced Research in VLSI*, Mar 1999, pp. 215–229.

[38] P.Hasler, B. Minch, and C. Diorio, "Floating-gate devices: they are not just for digital memories any more," in *IEEE International Symposium on Circuits and Systems*, vol. 2, Jul 1999, pp. 388–391 vol.2.

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

[40] T. Constandinou, J. Georgiou, and C. Toumazou, "An auto-input-offset removing floating gate pseudo-differential transconductor," in *Proceedings of the International Symposium on Circuits and Systems*, vol. 1, May 2003, pp. I–169–I–172 vol.1.

[41] S. Shah and S. Collins, "A temperature independent trimmable current source," in *IEEE International Symposium on Circuits and Systems*, vol. 1, 2002, pp. I–713–I–716 vol.1.

[42] S. A. Jackson, J. Killens, and B. Blalock, "A programmable current mirror for analog trimming using single poly floating-gate devices in standard cmos technology," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 1, pp. 100–102, Jan 2001.

[43] A. Negut and A. Manolescu, "Analog floating gate approach for programmable current mirrors and current sources," in *International Semiconductor Conference*, vol. 02, Oct 2010, pp. 525–528.

[44] T. Hall, C. Twigg, J. Gray, P. Hasler, and D. Anderson, "Large-scale field-programmable analog arrays for analog signal processing," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 11, pp. 2298–2307, nov 2005.

[45] R. B. Wunderlich, F. Adil, and P. Hasler, "Floating Gate-Based Field Programmable Mixed-Signal Array," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 8, pp. 1496–1505, August 2013.

[46] 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.

[47] P. Gray, P. Hurst, S. Lewis, and R. Meyer, *Analysis and Design of Analog Integrated Circuits*, 5th ed. Wiley, 2009.

[48] S. Song, K. C. Chun, and C. H. Kim, "A logic-compatible embedded flash memory for zero-standby power system-on-chips featuring a multi-story high voltage switch and a selective refresh scheme," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 5, pp. 1302–1314, May 2013.

[49] C. Diorio, "A p-channel mos synapse transistor with self-convergent memory writes," *IEEE Transactions on Electron Devices*, vol. 47, no. 2, pp. 464–472, Feb 2000.

[50] T. Ong, P. Ko, and C. Hu, "Hot-carrier current modeling and device degradation in surface-channel p-mosfets," *IEEE Transactions on Electron Devices*, vol. 37, no. 7, pp. 1658–1666, July 1990.

[51] 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, pp. 2685–2692, 2011.

[52] A. Bandyopadhyay, G. Serrano, and P. Hasler, “Adaptive algorithm using hot-electron injection for programming analog computational memory elements within 0.2 percent of accuracy over 3.5 decades,” *IEEE Journal of Solid-State Circuits*, vol. 41, pp. 2107–2114, 2006.

[53] A. Basu and P. E. Hasler, “A fully integrated architecture for fast and accurate programming of floating gates over six decades of current,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, pp. 953–962, 2011.

[54] A. Sedra and K. Smith, “A second-generation current conveyor and its applications,” *IEEE Transactions on Circuit Theory*, vol. 17, no. 1, pp. 132–134, February 1970.

[55] B. Rumberg and D. Graham, “A low-power and high-precision programmable analog filter bank,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 59, no. 4, pp. 234–238, April 2012.

[56] B. Rumberg, D. Graham, V. Kulathumani, and R. Fernandez, “Hibernets: Energy-efficient sensor networks using analog signal processing,” *IEEE J. Emerging and Sel. Topics Circuits and Syst.*, vol. 1, no. 3, pp. 321–334, Sept. 2011.

[57] A. Dilello, S. Andryzcik, B. M. Kelly, B. Rumberg, and D. W. Graham, “Temperature compensation of floating-gate transistors in field-programmable analog arrays,” in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2017, pp. 1–4.

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

[59] M. M. Navidi, D. W. Graham, and B. Rumberg, “Below-ground injection of floating-gate transistors for programmable analog circuits,” in *IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2017, pp. 1–4.

[60] B. Kelly and D. Graham, “An asynchronous adc with reconfigurable analog preprocessing,” in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2016, pp. 1062–1065.

[61] P. R. Kinget, “Scaling analog circuits into deep nanoscale cmos: Obstacles and ways to overcome them,” in *Proceedings of IEEE Custom Integrated Circuits Conference*, 2015, pp. 1–8.

[62] R. H. Walden, “Analog-to-digital converter survey and analysis,” *IEEE Journal on Selected Areas in Communications*, pp. 539–550, 1999.

[63] 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.

[64] 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.

[65] 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.

[66] 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.

[67] 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.

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

[69] J. W. Lee, J. Y. Lee, and J. J. Lee, “Jenga-inspired optimization algorithm for energy-efficient coverage of unstructured WSNs,” *IEEE Wireless Communications Letters*, vol. 2, no. 1, pp. 34–37, 2013.

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

[71] 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.

[72] 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.

[73] D. Schinkel, E. Mensink, E. Klumperink, E. van Tuijl, and B. Nauta, in *Proc. of IEEE ISSCC*, Feb. 2007, pp. 314–605.

[74] 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.

[75] 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.

[76] 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.

[77] G. Farin, *Curves and Surfaces for CAGD: A Practical Guide.* Academic Press, 1996.

[78] C. J. DeLuca, "The use of surface electromyography in biomechanics," *J. Appl. Biomech.*, pp. 135–163, 1997.

[79] 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.

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

[81] C. I. Ieong, P. I. Mak, C. P. Lam, C. Dong, M. I. Vai, and P. U. Mak, "A 0.83μw qrs detection processor using quadratic spline wavelet transform for wireless ecg acquisition in 0.35μm cmos," *IEEE Trans. Biomed. Circuits Syst*, pp. 586–595, 2012.

[82] 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 IEEE Custom Integrated Circuits Conference*, 2012, pp. 1–4.

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

[84] X. Zhang and Y. Lian, "A 300mv 220nw event-driven adc with real-time qrs detection for wearable ecg sensors," *IEEE Transactions on Biomedical Circuits and Systems*, pp. 834–843, 2014.

[85] S. Liu, J. Kramer, G. Indiveri, T. Delbrck, R. Douglas, and C. A. Mead, *Analysis and Synthesis of Static Translinear Circuits.* MITP, 2002, pp. 177–227. [Online]. Available: https://ieeexplore.ieee.org/document/6290146

[86] J. Chang, Y. Chen, W. Chan, S. P. Singh, H. Cheng, H. Fujiwara, J. Lin, K. Lin, J. Hung, R. Lee, H. Liao, J. Liaw, Q. Li, C. Lin, M. Chiang, and S. Wu, "12.1 a 7nm 256mb sram in high-k metal-gate finfet technology with write-assist circuitry for low-vminapplications," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, Feb 2017, pp. 206–207.

[87] *1MHz, Low-Power Op Amp*, Microchip Technology, 4 2009, rev. J.

Appendix A

Wheatstone Bridge Resistive Sensing Circuit

The output voltage expression for Fig. A.1 (shown here for clarity) was freely given in Chapter 2, but is derived here. Additional helper variables are included in Fig. A.1 for ease in explanation and will be referenced in the equations. The first assumption in this derivation is that the open-loop gain of the operational amplifiers (op-amps) is very high (i.e. $a_{ov} = 100,000$) such that the ideal op-amp characteristics can be applied for closed-loop configurations. The op-amps employed on the physical circuit for application were MCP6001 series, which meets this criteria and have a typical open-loop gain of 112dB [87].

Beginning at the output, we have the following basic expression where I_2 will need to be expressed in terms of the reference voltages and the thermistor resistance.

$$V_{out} = V_{ref2} + I_2 R_2 \tag{A.1}$$

$$I_2 = I_3 - I_1 = \frac{V_{ref2} - V_x}{R} - \frac{V_{ref1} - V_{ref2}}{R_1} \tag{A.2}$$

Then substituting A.2 into A.1 yields

$$V_{out} = V_{ref2} + R_2 \left[\frac{V_{ref2} - V_x}{R} - \frac{V_{ref1} - V_{ref2}}{R_1} \right] \tag{A.3}$$

An additional I_1 expression can be written from the left branch of the Wheatstone bridge from V_{ref2} to V_x.

$$I_1 = \frac{V_{ref2} - V_x}{R} \tag{A.4}$$

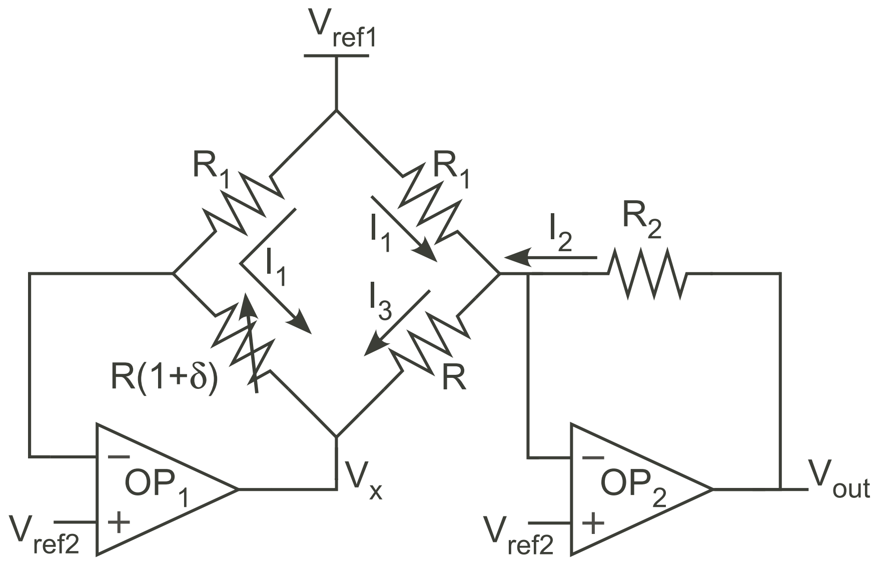


Figure A.1: Thermistor sensing circuit that utilizes two operational amplifiers and Wheatstone bridge. This designs is also capable of operating on a single supply rail like a battery source.

Then, re-arranging Eq. A.4 in terms of V_x yeilds the following.

$$V_x = V_{ref2} - I_1[R(1+\delta)] \tag{A.5}$$

Substituting A.4 into the previous expression gives the following.

$$V_x = V_{ref2} - \frac{V_{ref1} - V_{ref2}}{R_1}[R(1+\delta)] \tag{A.6}$$

Then, substituting the expression for V_x back into A.7.

$$V_{out} = V_{ref2} + R_2\left[\frac{V_{ref2}}{R} - \frac{V_{ref2} - R[1+\delta]\frac{V_{ref1}-V_{ref2}}{R_1}}{R} - \frac{V_{ref1} - V_{ref2}}{R_1}\right] \tag{A.7}$$

Rearranging the terms inside the brackets reveal that some terms can be cancelled.

$$V_{out} = V_{ref2} + R_2\left[\frac{\cancel{V_{ref2}}}{\cancel{R}} - \frac{\cancel{V_{ref2}}}{\cancel{R}} + \frac{\cancel{R}[1+\delta]\frac{V_{ref1}-V_{ref2}}{R_1}}{\cancel{R}} - \frac{V_{ref1} - V_{ref2}}{R_1}\right] \tag{A.8}$$

The $[1+\delta]$ term can be expanded allowing for additional terms to be cancelled.

$$V_{out} = V_{ref2} + R_2\left[\cancel{\frac{V_{ref1} - V_{ref2}}{R_1}} + \delta\frac{V_{ref1} - V_{ref2}}{R_1} - \cancel{\frac{V_{ref1} - V_{ref2}}{R_1}}\right] \tag{A.9}$$

This yields the final output equation expressing V_{out}.

$$V_{out} = V_{ref2} + \frac{R_2}{R_1}\delta(V_{ref1} - V_{ref2}) \tag{A.10}$$